การพัฒนาเวิร์คโฟลว์และเว็บเซอร์วิสที่เหมาะสมในการวิเคราะห์สนิป

Development of Suitable Workflows and Web Services for SNP Analysis

กษิดิ์กฤษณ์ ดำเกลี้ยง

Kasikrit Damkliang

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญา

วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Fulfillment of the Requirements for the Degree of

Master of Engineering in Computer Engineering

Prince of Songkla University

2552

ชื่อวิทยานิพนธ์ การพัฒนาเวิร์คโฟลว์และเว็บเซอร์วิสที่เหมาะสมในการวิเคราะห์สนิป
ผู้เขียน นายกษิดิ์กฤษณ์ ดำเกลี้ยง
สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

……
(ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

……
(ผู้ช่วยศาสตราจารย์ ดร.สุนทร วิทูสุรพจน์)

คณะกรรมการสอบ

…….....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

…….…….....กรรมการ
(รองศาสตราจารย์ ดร.วสันต์ จันทราทิตย์)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุนทร วิทูสุรพจน์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้สำหรับการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

……..……
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การพัฒนาเวิร์คโฟลว์และเว็บเซอร์วิสที่เหมาะสมในการวิเคราะห์สนิป
ผู้เขียน นายกษิดิ์กฤษณ์ ดำเกลี้ยง
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2552

บทคัดย่อ

วิทยานิพนธ์นี้ นำเสนอการพัฒนาเวิร์คโฟลว์เพื่อวิเคราะห์สนิปของกุ้งและมนุษย์ โดยใช้เทคโนโลยีมายกริด โปรแกรมทาเวอร์นา เว็บเซอร์วิส และนำเสนอกระบวนการที่เหมาะสมในการเลือกเว็บเซอร์วิสและวิธีการพัฒนาหรือการวาดเวิร์คโฟลว์ โดยใช้งานวิจัยการวิเคราะห์สนิปทั้งสองงานเป็นกรณีศึกษา

เวิร์คโฟลว์ในกรณีศึกษาเรื่องสนิปของกุ้ง มักเกิดปัญหาใช้เวลาเกินและได้ผลลัพธ์ไม่คงเส้นคงวาเมื่อจำนวนข้อมูลเกินขอบเขตที่บริการภายนอกรับได้ วิทยานิพนธ์นี้จึงนำเสนอการพัฒนาบริการท้องถิ่นขึ้นมาใช้งานเอง เพื่อแก้ไขปัญหาดังกล่าว

ในงานวิเคราะห์สนิปมนุษย์ เวิร์คโฟลว์มีความซับซ้อนและมีหลายชั้น วิทยานิพนธ์นี้นำเสนอกลไกการลดเวลาในการพัฒนาและทดสอบเวิร์คโฟลว์ โดยพัฒนาเวิร์คโฟลว์สำหรับตรวจสอบความถูกต้องของเวิร์คโฟลว์เป้าหมายก่อนการทำงานจริง ว่าทันสมัยพร้อมทำงานหรือล้าสมัยไปแล้วและให้ข้อมูลที่เกี่ยวข้องเพื่อการปรับแก้เวิร์คโฟลว์ ซึ่งสามารถลดเวลาที่เสียไปในการตรวจสอบและหาที่ผิดได้เป็นอย่างมาก

คำสำคัญ: เวิร์คโฟลว์, ทาเวอร์นา, เว็บเซอร์วิส, ไบโอมาร์ท, สนิป, การตรวจสอบความถูกต้อง

Thesis Title	Development of suitable workflows and Web services for SNP analysis
Author	Mr. Kasikrit Damkliang
Major Program	Computer Engineering
Academic Year	2009

ABSTRACT

This thesis presents workflow development for shrimp and human SNP analysis using myGrid technology, Taverna, and Web service technology. Through the two case studies of shrimp and human SNP analysis, this thesis suggests suitable processes in Web service selection and workflow composition.

The first workflow for shrimp SNP analysis sometimes fails due to time-out and inconsistency problems caused by external services when the number of data exceeds the service's limit. Our solution is providing local Web service interfaces instead.

In case of human SNP analysis, the workflow is complex and nested. We propose a time reduction mechanism in development and testing using a validation workflow. This validation workflow is to be used before running a workflow to show whether there are up-to-date or out-of-date. It also provides associate information for re-composing and configuring the workflow. Therefore, it can greatly reduce the debugging and checking times.

Keywords: Workflow, Taverna, Web Service, SNP, Validation

กิตติกรรมประกาศ

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.พิชญา ตัณฑัยย์ อาจารย์ที่ปรึกษาหลักที่ได้คำปรึกษาและความรู้ในด้านต่างๆ พร้อมทั้งชี้แนะแนวทางการดำเนินการเกี่ยวกับวิทยานิพนธ์ รวมถึงการให้โอกาสและประสบการณ์ในด้านวิชาการต่างๆทั้งในประเทศและต่างประเทศ อีกทั้งสนับสนุนอุปกรณ์ในการทำวิจัย ตลอดจนช่วยตรวจและแก้ไขปรับปรุงวิทยานิพนธ์ให้เป็นอย่างสมบูรณ์ และที่สำคัญคือการมีอาจารย์เป็นกัลยาณมิตรธรรม

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.สุนทร วิทูสุรพจน์ อาจารย์ที่ปรึกษาร่วม ที่ได้คอยชี้แนะแนวทางต่างๆที่เป็นประโยชน์ และรวมทั้งได้ให้ความรู้ความเข้าใจที่จำเป็นต่อการทำวิจัย

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ที่ได้ให้คำแนะนำในการปรับปรุงวิทยานิพนธ์ให้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณคณาจารย์และบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในระหว่างทำวิทยานิพนธ์

ขอขอบพระคุณ คณะวิศวกรรมศาสตร์ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ โครงการสร้างโครงสร้างพื้นฐานด้วยไบโอกริดภายใต้การสนับสนุนของศูนย์ไทยกริดแห่งชาติและสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (องค์การมหาชน) ศูนย์กริดแห่งมหาวิทยาลัยสงขลานครินทร์ และศูนย์ความเป็นเลิศด้านชีววิทยาศาสตร์ของประเทศไทย (TCELS) คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล ที่ได้มอบทุนสนับสนุนในการทำวิจัย

ขอขอบคุณนึกศึกษาปริญญาโทสาขาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้คำแนะนำและเป็นกำลังใจมาโดยตลอด

ขอขอบพระคุณ พ่อแม่และทุกคนในครอบครัว ผู้ให้กำลังใจและความหวังในการฝ่าฟันอุปสรรค และสนับสนุนให้การเรียนได้สำเร็จลุล่วงไปด้วยดี

ท้ายที่สุด ขอกราบขอบพระคุณพระอาจารย์ปราโมทย์ ปาโมชโช และครูบาอาจารย์ทุกท่านทั้งที่ยังดำรงขันธ์อยู่และได้ละขันธ์ไปแล้ว ในคำเทศนาธรรมและคำสอนอันเป็นทางเอก ที่ทำให้ผู้ทำวิทยานิพนธ์ได้ยืนหยัดและยอมรับไปตามความเป็นจริง

กษิดิ์กฤษณ์ ดำเกลี้ยง

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

รายการตาราง

รายการตาราง (ต่อ)

รายการรูป

รายการรูป (ต่อ)

รายการรูป (ต่อ)

รายการรูป (ต่อ)

รายการรูป (ต่อ)

สัญลักษณ์คำย่อและตัวย่อ

GWAS	Genome Wide Association Study
SNP	Single Nucleotide Polymorphism
SOA	Service Oriented Architecture
XML	Extensible Markup Language
HTTP	Hypertext Transfer Protocol
SOAP	Service Oriented Architecture Protocol
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery and Integration
EBI	European Bioinformatics Institute
SCUFL	Simple Conceptual Unified Flow Language
WABI	Web API for Biology
DDBJ	DNA Data Bank of Japan
BLAST	Basic Local Alignment Search Tool
AME	Advanced Model Explorer
LGPL	GNU Lesser General Public License

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของวิทยานิพนธ์

ในอดีตที่ผ่านมานักวิจัยด้านชีวสารสนเทศศาสตร์ ต้องทำงานในลักษณะที่เรียกว่า *"คัดลอกแล้ววาง"* (Copying and Pasting) กล่าวคือการคัดลอกข้อมูลที่ต้องการนำไปทำงานต่อจากหน้าเว็บหนึ่ง แล้วนำไปวางเพื่อเป็นอินพุตในอีกหน้าเว็บหนึ่ง โดยใช้หลายๆเว็บหรือหลายโปรแกรมเพื่อให้ได้ผลลัพธ์ที่ต้องการ ซึ่งวิธีการแบบนี้ ต้องใช้เวลามาก เสี่ยงต่อความผิดพลาด และไม่สามารถทำงานวิจัยที่ใหญ่ขึ้นได้ เนื่องจากกับความซับซ้อนของงานวิจัยและการไม่มีมาตรฐานร่วมกันของข้อมูล ดังนั้นระบบการทำงานแบบเวิร์คโฟลว์จึงถูกพัฒนาขึ้น เพื่อช่วยในการสร้างเวิร์คโฟลว์ของระบบงาน เป็นการจัดการการเข้าถึงทรัพยากรต่างๆที่กระจัดกระจายในอินเตอร์เน็ตด้วยเวิร์คโฟลว์และความสามารถในการจัดการแปลงข้อมูลระหว่างมาตรฐานต่างๆ [1][2][3]

โปรแกรมทาเวอร์นา (Taverna) พัฒนาขึ้นในโครงการมายกริด (myGrid) [3][4] ซึ่งเป็นโปรแกรมประยุกต์ที่รวบรวมการเข้าถึงบริการและทรัพยากร ที่กระจายอยู่ในอินเตอร์เน็ตเหล่านั้นเข้าไว้ด้วยกัน เป็นเสมือนหนึ่งว่าอยู่บนระบบเดียวกันด้วยเทคโนโลยีเว็บเซอร์วิส (Web Service) อย่างไรก็ตามงานวิจัยที่ใช้เทคโนโลยีเวิร์คโฟลว์ดังกล่าว ยังจัดว่ามีลักษณะเฉพาะตัวจึงจำเป็นต้องพัฒนาเวิร์คโฟลว์ที่เหมาะสมและพัฒนาเว็บเซอร์วิสเพิ่มเติม เพื่อให้ทำงานได้อย่างถูกต้อง ให้ผลลัพธ์ที่ตรงตามวัตถุประสงค์และมีประสิทธิภาพในการทำงาน

การพัฒนาเวิร์คโฟลว์เชื่อมโยงเว็บเซอร์วิส เพื่อวิเคราะห์และทำนายผลกระทบของสนิป (Single Nucleotide Polymorphism หรือ SNP) ในงานวิจัยเภสัชพันธุศาสตร์ เป็นโครงการที่ร่วมมือกับนักชีวสารสนเทศของหน่วยไวรัสวิทยาและจุลชีววิทยาโมเลกุล ภาควิชาพยาธิวิทยา คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล โดยงานวิจัยในลักษณะนี้เรียกว่า Genome Wide Association Study หรือ GWAS [5] เป็นงานวิจัยที่ซับซ้อนมากไม่สามารถทำให้สำเร็จได้อย่างมีประสิทธิภาพและรวดเร็ว ด้วยวิธีการตัดต่อข้อมูลจากอินเตอร์เน็ตโดยวิธีคัดลอกแล้ววางโดยนักวิจัย (Manual) เพียงอย่างเดียวเท่านั้น เนื่องจากข้อมูลสนิป [6] มีจำนวนมากมายถึง

ประมาณ 10 ล้านตำแหน่ง กระจัดกระจายกันอยู่บนโครโมโซม 23 คู่ของมนุษย์ โดยสนิปจะตอบสนองต่อการใช้ยาและภาวะการที่เป็นโรค [6][7] และข้อมูลที่ใช้ก็กระจัดกระจายอยู่ที่ต่างๆบนอินเตอร์เน็ต ซึ่งอาจจะมีโอกาสผิดพลาดได้สูงหากให้นักวิจัยตัดต่อข้อมูลด้วยวิธีคัดลอกแล้ววางดังกล่าว ดังนั้นกรณีศึกษาคือการพัฒนาเวิร์คโฟลว์โดยใช้โปรแกรมทาเวอร์นา ในการทำนายผลกระทบอันเนื่องมาจากการเปลี่ยนแปลงของลำดับเบส (SNP) ต่างๆ รวมทั้งสร้างระบบที่มีความสามารถในการคัดเลือกตำแหน่งสนิป ที่น่าสนใจในการนำไปศึกษาด้านหน้าที่การทำงาน (Functional Study) หรือผลกระทบต่อโปรตีนของงานด้านชีวสารสนเทศศาสตร์ต่อไป

นอกจากนี้การใช้ฐานข้อมูลจากหลายๆที่ช่วยเลือกยีน หรือจัดลำดับยีนที่มีความน่าสนใจหรือมีความสำคัญเพื่อให้สามารถนำไปศึกษาต่อได้ ซึ่งจะเป็นเรื่องที่ยากมากหากไม่ใช้เวิร์คโฟลว์ เพราะข้อมูลมีขนาดใหญ่และใช้จากหลายๆแหล่งที่กระจัดกระจายกันอยู่บนอินเตอร์เน็ต การเลือกเว็บเซอร์วิสที่ถูกต้องและตรงตามจุดประสงค์มาใช้ประกอบเป็นเวิร์คโฟลว์นั้น ไม่สามารถจะเลือกได้ทันที เนื่องจากปัจจุบันมีบริการทางด้านชีวสารสนเทศศาสตร์กว่า 30,000 บริการ ซึ่งทำหน้าที่ทั้งที่เหมือนกันและแตกต่างกันไปตามแต่ละบริการนั้นๆ [8][9] ดังนั้นวิทยานิพนธ์นี้จะเกี่ยวข้องกับการวาดเวิร์คโฟลว์และการเลือกเว็บเซอร์วิสที่เหมาะสม โดยให้ข้อเสนอแนะที่เป็นรูปธรรมซึ่งเป็นวัตถุประสงค์แรก

การทดลองสร้าง และการทำงานกับเวิร์คโฟลว์กรณีศึกษาการวิเคราะห์สนิปของมนุษย์ด้านเภสัชพันธุศาสตร์ที่เป็นงานวิจัยร่วมกับมหาวิทยาลัยมหิดล บริการหลักที่เลือกนำมาใช้งานคือบริการไบโอมาร์ท (BioMart) [11][12] และเนื่องจากงานวิจัยมีความซับซ้อนจึงพัฒนาเวิร์คโฟลว์ทีละส่วนๆตามลักษณะขั้นตอนของการทำงานขั้นตอนนั้นๆ นอกจากนี้เวิร์คโฟลว์หนึ่งๆมีการทำงานในลักษณะ Data flow กล่าวคือ เอาท์พุตของบริการหนึ่งๆ จะเป็นอินพุตให้กับบริการอื่นๆ หากบริการใดบริการหนึ่งผลิตผลลัพธ์ไม่ถูกต้องตรงกับวัตถุประสงค์ของการทำงาน ก็จะทำให้ Data flow ในเวิร์คโฟลว์ส่วนนั้นไม่ถูกต้องตรงตามวัตถุประสงค์ไปด้วย และโปรแกรมทาเวอร์นาเองก็ไม่มีวิธีการดีบักหรือตรวจสอบใดๆเพียงพอที่จะตรวจสอบเวิร์คโฟลว์เพื่อแก้ปัญหานี้ได้ จากการทดลองสร้างและทำงานกับเวิร์คโฟลว์ในงานวิจัยที่กล่าวข้างต้น สามารถสรุปสาเหตุที่ทำให้เสียเวลาได้ดังนี้

1) หากเวิร์คโฟลว์ใดใช้เวลาในการทำงานที่ยาวนาน ก็ต้องรอจนกว่าเวิร์คโฟลว์นั้นๆจะทำงานเสร็จสิ้นเสียก่อน จึงจะทราบว่าผลลัพธ์การทำงานนั้นไม่ถูกต้องตามความต้องการ ทำให้เสียเวลามาก เช่น เป็นชั่วโมงๆหรือเป็นวัน เป็นต้น

2) เมื่อทราบว่าผลลัพธ์การทำงานไม่ถูกต้องตามข้อที่ 1 แล้ว ผู้ใช้งานก็จะเริ่มการดีบัก (Debug) หรือตรวจสอบ (Validate) Data flow ของแต่ละบริการเพื่อให้ทราบว่าบริการหรือ โปรเซสเซอร์ (Processor) ใดในเวิร์คโฟลว์ที่ทำงานผิดพลาด

3) เมื่อทราบแล้วว่าบริการ (Processor) ใดทำงานผิดพลาด ผู้ใช้งานจะพยายามตีกรอบการตรวจสอบให้แคบลงโดยการตรวจสอบอย่างถี่ถ้วนที่บริการนั้นๆ ไม่ว่าจะเป็นชนิดของข้อมูลหรือไวยากรณ์ของสคริปต์ต่างๆ ซึ่งในบางครั้งก็ถูกต้องดีแล้ว แต่เมื่อทดสอบโดยให้เวิร์ค-โฟลว์ทำงานอีกครั้ง ก็ยังให้ผลผิดพลาดเช่นเดิม

4) เมื่อเป็นดังข้อที่ 1 ถึง 3 ก็พอที่จะสรุปได้ว่า ปัญหาอาจจะไม่ได้เกิดกับ Data flow แต่น่าจะอยู่ที่ตัวบริการเองต่างหาก ดังนั้นผู้ใช้จะเข้าไปตรวจสอบข้อกำหนดการปรับแต่งของบริการไบโอมาร์ทที่เลือกใช้ (Configuration) ซึ่งก็จะพบว่าคุณลักษณะ (ฟิลเตอร์และแอ็ตตริบิวต์) บางค่าของบริการที่เรียกใช้ในเวิร์คไฟลว์นั้น ได้ล้าสมัยหรือไม่มีให้บริการแล้ว

5) วิธีการตรวจสอบของโปรแกรมทาเวอร์นาเอง ก็ไม่เพียงพอที่จะตรวจสอบเวิร์คโฟลว์เพื่อแก้ปัญหานี้ได้ ทำให้ต้องเสียเวลาไปมากเป็นหลายๆชั่วโมงหรือเป็นวัน ยิ่งเวิร์คโฟลว์มีความซับซ้อนทั้งจำนวนบริการไบโอมาร์ทและจำนวนคุณสมบัติมากเท่าไร ก็ยิ่งเสียเวลาเพิ่มขึ้นไปด้วย

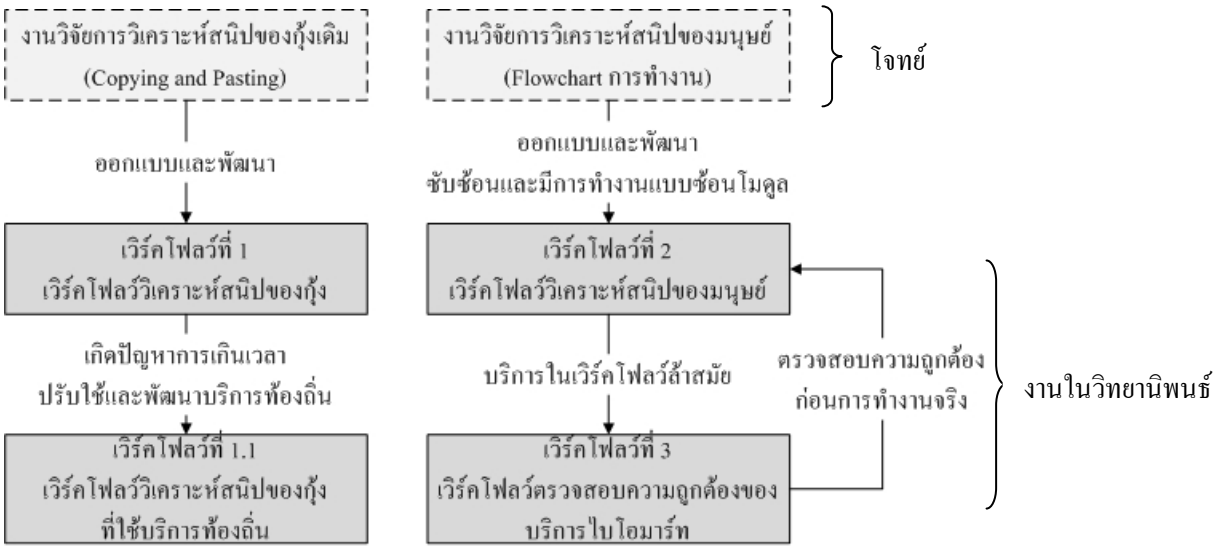
6) สำหรับการแก้ปัญหา Data flow ที่มีความผิดพลาดคือ การวาดเวิร์คโฟลว์ใหม่ โดยการลบบริการที่ล้าสมัยออก และเพิ่มบริการเดิมที่เป็นปัจจุบันลงไปในเวิร์คโฟลว์ไดอะแกรมที่มีปัญหาดังกล่าว โดยเลือกจากรายชื่อบริการที่ปรากฏในโปรแกรมทาเวอร์นา เพราะการเริ่มต้นของโปรแกรมทาเวอร์นาทุกครั้งนั้น จะมีการค้นหาบริการต่างๆจากข้อมูลที่เป็นค่าโดยปริยายของโปรแกรมเอง (Default) และไบโอมาร์ทก็เป็นส่วนหนึ่งของค่าโดยปริยายนั้นแล้ว ดังนั้นบริการต่างๆที่ปรากฏในรายชื่อบริการ จะเป็นบริการที่ทันสมัยที่สุดที่โปรแกรมนำเสนอต่อผู้ใช้งานในขณะเวลานั้น[8][9]

การค้นหาบริการไบโอมาร์ทใดๆที่มีคุณสมบัติต่างๆที่ล้าสมัย หรือสามารถกล่าวได้ว่าบริการไบโอมาร์ทที่ล้าสมัยนั้น จะต้องทำด้วยวิธีการตรงไปตรงมาโดยผู้ใช้งานเอง (Manual) คือ ผู้ใช้ต้องเรียกหน้าต่างการปรับแต่งของบริการไบโอมาร์ทขึ้นมา แล้วสังเกตดูว่าคุณสมบัติใดที่เลือกไว้ก่อนหน้านั้น ยังปรากฏให้เลือกได้ในหน้าต่างการปรับแต่งนี้หรือไม่ หากไม่มีแล้ว แสดงว่าคุณสมบัตินั้นๆได้ถูกผู้ให้บริการเอาออกหรือเปลี่ยนไปเป็นชื่อใหม่ เป็นต้น เหล่านี้ล้วนต้องอาศัยประสบการณ์ และการสังเกตจากการใช้บริการไบโอมาร์ทของผู้ใช้งานเอง ยิ่งกว่านั้น จำนวนของบริการไบโอมาร์ทที่อยู่ในเวิร์คโฟลว์, จำนวนคุณสมบัติที่เรียกใช้ในบริการไบโอมาร์ท และความ

ซับซ้อนของเวิร์คโฟลว์ ปัจจัยเหล่านี้ ล้วนมีผลต่อเวลาในการตรวจสอบหรือการค้นหาสาเหตุที่ทำให้เวิร์คโฟลว์ทำงานได้ผลลัพธ์ที่ไม่ตรงกับความต้องการ

ดังนั้นวิทยานิพนธ์นี้ จึงเสนอการพัฒนากลไกการตรวจสอบเวิร์คโฟลว์ก่อนการทำงานโดยจะมุ่งเน้นไปที่การตรวจสอบบริการไบโอมาร์ทในเวิร์คโฟลว์วิเคราะห์สนิป เพราะเป็นบริการหลักที่ใช้ในเวิร์คโฟลว์ โดยเสนอการพัฒนาเวิร์คโฟลว์ภายใต้สิ่งแวดล้อมของโปรแกรมทาเวอร์นาที่สามารถตรวจสอบคุณสมบัติของบริการไบโอมาร์ทได้ เพื่อให้ข้อมูลที่มีนัยสำคัญต่อผู้ใช้งาน จะได้มีความสะดวกและง่ายต่อการตรวจสอบ และปรับแก้เวิร์คโฟลว์ซึ่งสามารถลดเวลาที่เสียไปได้อย่างมากเป็นวัตถุประสงค์ที่สอง

อีกประการหนึ่ง ในการสร้างเวิร์คโฟลว์เพื่อวิเคราะห์การเปลี่ยนแปลงลำดับเบส (Single-nucleotide polymorphism หรือ SNP) [6] ในกุ้ง เพื่อประโยชน์ในการพัฒนาสายพันธุ์ซึ่งเป็นความร่วมมือกับสถานวิจัยจีโนมิกส์และชีวสารสนเทศ มหาวิทยาสงขลานครินทร์ พบว่าในกระบวนการวิเคราะห์ข้อมูลลำดับพันธุกรรมของกุ้ง เพื่อสร้างโครงสร้างต้นไม้สายวิวัฒนาการ (Phylogenetic tree) [13][15] มีการวิเคราะห์และถ่ายโอนข้อมูลจำนวนมากในแต่ละบริการ การทำงานมีความล่าช้าจนทำให้มีปัญหาการเกินเวลา (Time-out) และความไม่คงเส้นคงวาในการทำงานเกิดขึ้น (Inconsistency) ส่งผลให้เวิร์คโฟลว์ทำงานไม่สำเร็จส่งผลให้ไม่ได้ผลลัพธ์ที่ต้องการ จึงเสนอการสร้างบริการท้องถิ่นในส่วนการทำงานเพื่อสร้างโครงสร้างต้นไม้สายวิวัฒนาการ ในการแก้ไขปัญหาดังกล่าว ซึ่งสามารถช่วยประหยัดเวลาในประมวลผลและให้ผลลัพธ์ที่ถูกต้องตรงตามความต้องได้ ซึ่งเป็นส่วนหนึ่งของวิทยานิพนธ์นี้ด้วยในวัตถุประสงค์สุดท้าย



รูปที่ 1.1 สรุปงานในวิทยานิพนธ์

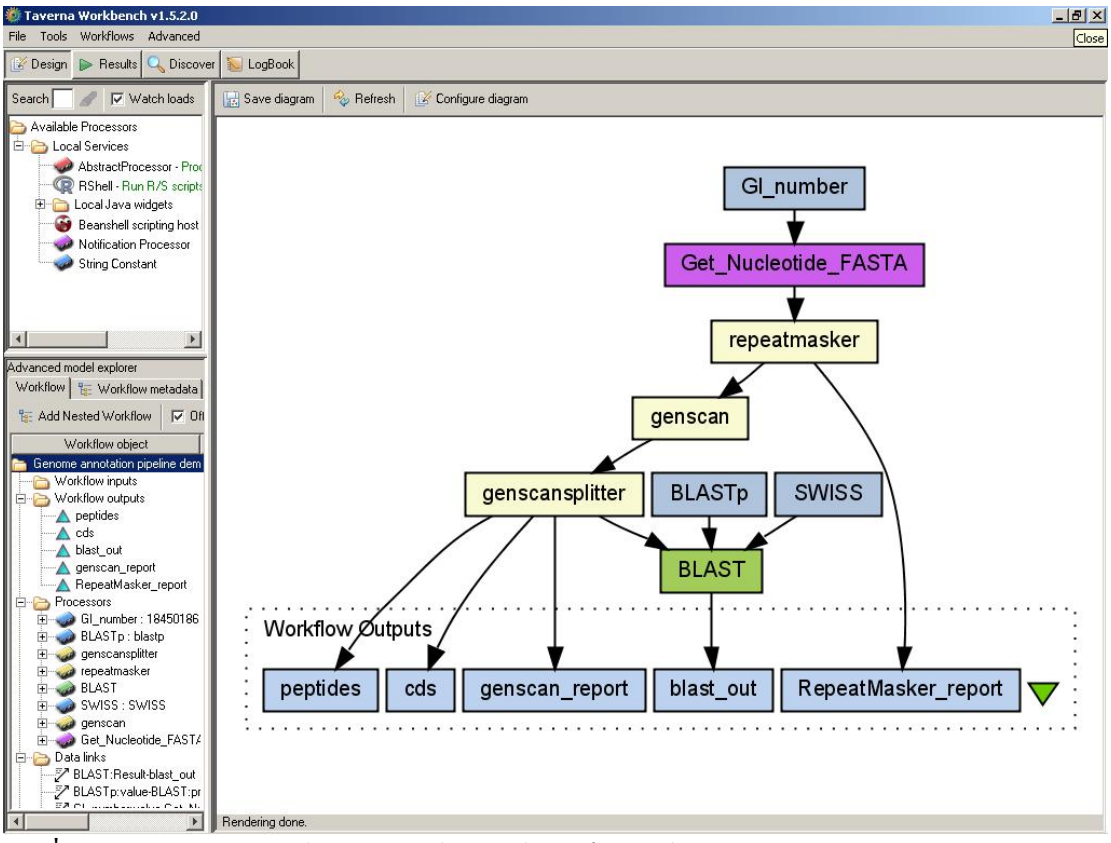
รูปที่ 1.1 แสดงแผนผังสรุปงานวิทยานิพนธ์ โดยประกอบด้วยการวิเคราะห์สนิปของกุ้งและของมนุษย์ วิทยานิพนธ์นี้ออกแบบพัฒนาเวิร์คโฟลว์เพื่อวิเคราะห์ของกุ้งได้เป็นเวิร์คโฟลว์ที่ 1 ซึ่งจากการทดสอบการทำงาน พบว่าเวิร์คโฟลว์เกิดปัญหาเกินเวลาและความไม่คงเส้นคงวาในการทำงาน จึงมีการแก้ไขปัญหาดังกล่าว โดยสร้างบริการท้องถิ่นขึ้นมาใช้งานเองในส่วนของกระบวนการทีเกิดปัญหาเท่านั้นและพัฒนาได้เป็นเวิร์คโฟลว์ที่ 1.1 ซึ่งสามารถแก้ไขปัญหาดังกล่าวได้

การวิเคราะห์สนิปของมนุษย์ มีระบบการทำงานความซับซ้อนและมีลักษณะการงานแบบซ้อนโมดูลๆหรือ Nested workflow วิทยานิพนธ์นี้ออกแบบและพัฒนาเวิร์คโฟลว์เพื่อวิเคราะห์ของมนุษย์ได้เป็นเวิร์คโฟลว์ที่ 2 แต่พบว่าเวิร์คโฟลว์เกิดปัญหาการล้าสมัยของบริการและทำงานได้ไม่ตรงตามความต้องการ จึงมีการแก้ไขปัญหาดังกล่าวโดยสร้างระบบตรวจสอบเวิร์ค-โฟลว์ก่อนการทำงานจริงได้เป็นเวิร์คโฟลว์ที่ 3 ซึ่งสามารถแก้ไขปัญหาดังกล่าวได้

1.2 การตรวจเอกสาร (Literatures Review)

1.2.1 โปรแกรมทาเวอร์นา

การทดลองแบบ *in silico* หรือการทำการทดลองด้วยคอมพิวเตอร์ในงานด้านวิทยาศาสตร์สุขภาพ (Life Sciences) เป็นการรวบรวมเครื่องมือวิเคราะห์และฐานข้อมูลที่กระจายในที่ต่างๆเข้าไว้ด้วยกัน การทดลองแบบนี้ทำได้โดยการ *คัดลอกแล้ววาง* ระหว่างเว็บเพจหรือโดยการเขียนโปรแกรมเพื่อปะติดปะต่อทรัพยากรเหล่านี้เข้าไว้ด้วยกันเช่น สคริปต์ของภาษา Perl เป็นต้น วิธีการเหล่านี้มีความจำเป็นมากขึ้นในงานชีวสารสนเทศ กลุ่มงานวิจัยอิสระต่างๆ (Autonomous research) [3][4] ได้สร้างและดูแลฐานข้อมูลที่แตกต่างกันและได้พัฒนาอัลกอริทึมและเครื่องมือวิเคราะห์ที่มีความสัมพันธ์กัน ทรัพยากรและปริมาณของข้อมูลสาธารณะที่เปิดให้เข้าใช้ได้เพิ่มขึ้นอย่างรวดเร็วในยุคข้อมูลจีโนม ปัจจัยเหล่านี้ทำให้เกิดการรวมตัวกันของกลุ่มวิจัยดังกล่าว อย่างไรก็ตามยังมีปัญหาเรื่องมาตรฐานการใช้ข้อมูลร่วมกัน ความแตกต่างกันของระบบที่มีอยู่เดิม และลักษณะทางด้านภูมิศาสตร์ที่ตั้งของบริการข้อมูลในอินเตอร์เน็ต



รูปที่ 1.2 โปรแกรมทาเวอร์นาแสดงเวิร์คโฟลว์และเว็บเซอร์วิสโดยการแบ่งประเภทด้วยสีต่างๆ [4]

โปรแกรมทาเวอร์นา (Taverna) (รูปที่ 1.2) พัฒนาขึ้นในโครงการมายกริดซึ่งเป็นโปรแกรมประยุกต์ที่รวบรวมการเข้าถึงบริการและทรัพยากร ที่กระจายอยู่ในอินเตอร์เน็ตเข้าไว้เป็นเสมือนหนึ่งว่าอยู่บนระบบเดียวกันด้วยเทคโนโลยีเว็บเซอร์วิส อย่างไรก็ตามงานวิจัยที่ใช้เทคโนโลยีเวิร์คโฟลว์ดังกล่าว ยังมีลักษณะเฉพาะตัวที่ไม่สามารถใช้เวิร์คโฟลว์ตามบริการตัวอย่างเวิร์คโฟลว์ที่มีให้ จึงจำเป็นต้องพัฒนาเวิร์คโฟลว์ที่เหมาะสม และพัฒนาเว็บเซอร์วิสเพิ่มเติม เพื่อให้ทำงานได้อย่างถูกต้อง ให้ผลลัพธ์ที่ตรงตามวัตถุประสงค์และมีประสิทธิภาพในการทำงาน [2][10]

1.2.2 โปรแกรมทาเวอร์นาสอง รุ่นทดลองใช้

โปรแกรมทาเวอร์นาตั้งแต่รุ่น 1.7.0 มีปลักอินชื่อว่า *Taverna 2 preview* ซึ่งเป็นการทดลองโปรแกรมทาเวอร์นารุ่น 2 เพื่อการทำงานที่เรียกว่า *Workflow health check* หรือการตรวจสอบเสถียรภาพของเวิร์คโฟลว์ โดยการตรวจสอบว่า เอ็นพอยต์ของเว็บเซอร์วิสต่างๆในเวิร์ค

โฟลว์มีตอบกลับผ่าน HTTP (Responding) หรือไม่ [8][9] หากมีการตอบกลับก็คือสามารถเข้าถึงได้ หากไม่มีการตอบกลับ ก็ไม่สามารถที่จะเปิดเวิร์คโฟลว์ขึ้นมาทำงานได้ ปัญหาก็อาจจะมาจากเครือข่ายของผู้ใช้งานเองไม่สามารถติดต่อภายนอกได้ หรือไม่ก็มีปัญหาที่ฝั่งของผู้ให้บริการเองอันเนื่องมาจากหลายๆสาเหตุ

อีกประการหนึ่งในการออกแบบและการสร้างเวิร์คโฟลว์ ในงานวิจัยด้านเภสัช-พันธุศาสตร์ซึ่งเป็นงานวิจัยที่มีขนาดใหญ่และมีกระบวนการที่ซับซ้อน จำเป็นต้องออกแบบการทำงานเป็นโมดูลๆทำงานในกระบวนการที่เกี่ยวข้องซึ่งมีการเรียกใช้โมดูลอื่นๆด้วย ในสิ่งแวดล้อมของทาเวอร์นาเรียกโมดูลการทำงานของแต่ละกระบวนเหล่านี้ว่า Nested workflow และพบว่า *Taverna 2 preview* ไม่สามารถตรวจสอบ Nested workflow ได้

ดังนั้น *Taverna 2 preview* ไม่สามารถตรวจจับบริการที่ล้าสมัยไปแล้วและไม่สามารถตรวจสอบความเสถียรภาพของเวิร์คโฟลว์ที่มีความซับซ้อนได้ การแก้ไขข้อจำกัดของ *Taverna 2 preview* เหล่านี้ เป็นส่วนหนึ่งของวิทยานิพนธ์นี้ด้วย

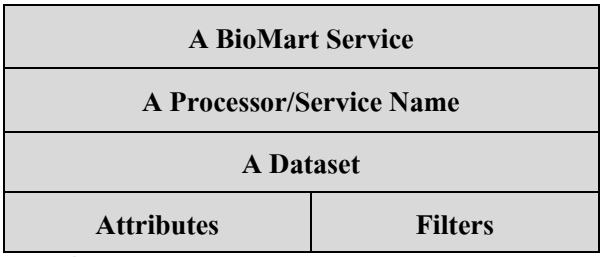
1.2.3 บริการไบโอมาร์ท (BioMart service)

ไบโอมาร์ท (BioMart) [11][12] คือระบบการจัดการข้อมูลเชิงคิวรีที่พัฒนาโดยสถาบันวิจัยมะเร็ง Ontario (OiCR) และสถาบันชีวสารเทศแห่งยุโรป (EBI) ระบบไบโอมาร์ทมีข้อมูลหลากหลายชนิดที่นำเสนอต่อผู้ใช้งาน เหมือนระบบเหมืองข้อมูลซึ่งทำหน้าที่ในการค้นหาข้อมูลได้อย่างซับซ้อน โดยมีหน้าต่างการทำงานเหมือนหน้าเว็บไซต์ที่เรียกว่า *'Out of the box'* ที่สามารถติดตั้งและปรับแต่งบริการได้ตามความต้องการของผู้ใช้งาน กลไกเหล่านี้มีลักษณะแบบกราฟิกส์และข้อความโดยมีพื้นฐานการทำงานแบบโปรแกรมประยุกต์ หรือใช้เทคโนโลยีเว็บเซอร์วิสหรือ Application Programming Interface (API) ที่เขียนด้วยภาษา Perl และ Java

ไบโอมาร์ทสร้างขึ้นเพื่อสนับสนุนการคิวรีข้อมูลได้เป็นอย่างดี และนอกจากนี้ยังสามารถปรับแต่งให้ทำงานกับเซิร์ฟเวอร์ฐานข้อมูล Distributed Annotation System (DAS) 1.5 ซึ่งใช้ในการแลกเปลี่ยนคำจำกัดความประกอบของลำดับจีโนมหรือโปรตีนได้ด้วย [16][17] โดยกระบวนการการแปลงจากแหล่งข้อมูลไปเป็นรูปแบบของไบโอมาร์ทนั้น สามารถทำได้อย่างอัตโนมัติด้วยเครื่องมือที่ไบโอมาร์ทได้เตรียมไว้ให้แล้ว ซึ่งสนับสนุนระบบจัดการฐานข้อมูลเชิงความสัมพันธ์ทั้ง MySQL [18], Oracle [19] และ Postgres [20] ไบโอมาร์ทเป็นซอตฟ์แวร์แบบเปิด

ภายใต้ลิขสิทธิ์ของ GNU Lesser General Public License (LGPL) [21] ซึ่งผู้ใช้สามารถเข้าถึงและใช้งานโดยไม่มีข้อผูกมัดหรือเงื่อนไขใดๆ

บริการไบโอมาร์ทหนึ่งๆจะประกอบด้วยชื่อของบริการ (Service Name) และเนื่องจากบริการไบโอมาร์ททำงานในลักษณะเหมืองข้อมูล ดังนั้นบริการของไบโอมาร์ทสามารถคิวรีข้อมูลจากดาตาเซ็ต (Dataset) ที่ผู้ให้บริการกำหนดไว้ได้ โดยในแต่ละดาตาเซ็ตก็จะประกอบด้วยคุณสมบัติต่างๆที่แตกต่างกันออกไปคือ ฟิลเตอร์ หรือฟิลด์ข้อมูลสำหรับเป็นเงื่อนไขการคิวรี (Filter or query filed) และแอ็ตตริบิวต์ หรือฟิลด์ข้อมูลสำหรับเป็นผลลัพธ์ของการคิวรี (Attribute or output filed) [22] ในการออกแบบเวิร์คโฟลว์เพื่อวิเคราะห์สนิปของมนุษย์พบว่าบริการไบโอมาร์ทหนึ่งๆสามารถให้บริการข้อมูลจากฐานข้อมูลเพียงหนึ่งฐานข้อมูล อย่างพอเพียงต่อความต้องการ และการทำงานแบบหนึ่งบริการต่อหนึ่งฐานข้อมูล ยังสะดวกต่อการพัฒนาและการปรับปรุงเวิร์คโฟลว์ได้ในอนาคต ดังนั้นงานวิจัยนี้จึงได้ตีกรอบเนื้อหาของบริการไบโอมาร์ทไว้ว่า บริการหนึ่งๆจะมีหนึ่งชื่อบริการและสามารถคิวรีข้อมูลได้หนึ่งฐานข้อมูล ซึ่งฐานข้อมูลหนึ่งๆประกอบด้วยหลายๆคุณสมบัติ ดังแสดงโครงสร้างบริการไบโอมาร์ทในรูปที่ 1.3



รูปที่ 1.3 โครงสร้างบริการไบโอมาร์ทในงานวิจัย

บริการไบโอมาร์ทเป็นปลักอินตัวหนึ่งในโปรแกรมทาเวอร์นา ทำให้สามารถใช้บริการข้อมูลได้อย่างง่ายด้วยเทคโนโลยีเว็บเซอร์วิส การเข้าถึงข้อมูลต่างๆจะเป็นลักษณะการสื่อสาร Java Database Connectivity (JDBC) [23] ซึ่งเป็น API ใน Java ที่ใช้สำหรับติดต่อไปยังฐานข้อมูลกลางของไบโอมาร์ทและนำเสนอต่อผู้ใช้งานในลักษณะของกราฟิกส์ สามารถติดตั้งและปรับแต่งบริการได้ตามความต้องการของผู้ใช้งาน [2]

1.2.4 การพัฒนาเวิร์คโฟลว์สำหรับการวิเคราะห์สนิปของมนุษย์

หน่วยไวรัสวิทยาและจุลชีววิทยาโมเลกุล คณะแพทยศาสตร์ โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล ศึกษาสนิปที่เกี่ยวข้องกับการแพ้ยาเนวิราฟีน (Nevirapine) [24] ซึ่งเป็นยาต้านเชื้อไวรัสเอชไอวี โดยเป็นยาในกลุ่ม Non-nucleoside reverse transcriptase inhibitor (NNRTI) [24] การแพ้ยาชนิดนี้จะมีอาการตั้งแต่เป็นผื่นคันเล็กน้อยจนถึงอาการหนัก และอาจทำให้เกิดโรค Steven's Johnson Syndrome [25] ซึ่งจะทำให้เกิดความลำบากในการใช้ชีวิตกับตัวผู้ติดเชื้อ และยังทำให้เสียเงินจำนวนมากมายในการรักษาอาการแพ้ยา ในท้ายที่สุดการแพ้ยาอาจทำให้ถึงแก่ชีวิตได้ การตรวจการเปลี่ยนแปลงของลำดับเบสหรือสนิป (SNP) ที่มีความเกี่ยวข้องหรือสัมพันธ์กับกลไกการแพ้ยา จะเป็นงานบริการที่สำคัญในอนาคตกับผู้ติดเชื้อเอชไอวีที่ต้องรับประทานยา Nevirapine โดยการศึกษาสนิปที่เกี่ยวข้องกับการแพ้ยา Nevirapine นี้จะใช้วิธีการทำแบบ GWAS มีกลุ่มการศึกษาแบ่งเป็น 2 กลุ่ม คือ กลุ่มที่ติดเชื้อเอชไอวีและได้รับประทานยาเนวิราฟีนแต่ไม่เกิดอาการแพ้ยา และกลุ่มที่ติดเชื้อเอชไอวีและได้รับประทานยาเนวิราฟีนและมีอาการแพ้ยาเกิดขึ้น ขณะนี้การศึกษาอยู่ในขั้นตอนของการวิเคราะห์ผลทางสถิติจากห้องปฏิบัติการ คาดว่าหลังจากที่วิเคราะห์ผลทางสถิติจากห้องปฏิบัติการเสร็จแล้ว จะยังคงมีตำแหน่งสนิปที่ต้องทำการศึกษาลงลึกไปในรายละเอียดต่อไปเพื่อหาตำแหน่งสนิปที่แท้จริง รวมทั้งกลไกที่เกี่ยวข้องกับการแพ้ยาเนวิ-ราฟีนนี้

Lude Franke และคณะ [26] ได้สร้างโปรแกรมคอมพิวเตอร์ที่มีชื่อว่า โปรแกรม Prioritizer โดยเป็นโปรแกรมที่ใช้ในการวิเคราะห์ผล Positional candidate gene prioritization โดยโปรแกรมจะทำหน้าที่นำตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของยีน หรือส่วนหนึ่งส่วนใดบนโครโมโซมที่สนใจ ไปเปรียบเทียบกับฐานข้อมูลต่างๆที่มีอยู่ในตัวโปรแกรม ประกอบด้วยฐานข้อมูล Gene ontology, Protein-Protein interaction และ Microarray co-expression และแสดงผลออกมาให้ผู้ใช้งานเห็นถึงความสัมพันธ์ที่เกี่ยวข้องกับข้อมูลที่ได้ใส่เข้าไปในระบบ วิธีการนี้จะทำให้สามารถคัดกรองจำนวนยีนหรือกลุ่มของยีนที่น่าจะนำไปศึกษาก่อนเป็นลำดับแรกได้ ในปัจจุบันทางผู้สร้างโปรแกรมได้นำเอาข้อมูลทางด้านสนิปและ Genetic linkage จากการสแกนจีโนมทั้งหมด (Whole genome scan) มาประกอบเข้าด้วยกันกับตัวโปรแกรมเดิม เพื่อเพิ่มขีดความสามารถในการคัดกรองตำแหน่งของสนิป หรือยีนที่เกี่ยวข้องกับการเกิดโรคหรือการแพ้ยาชนิดต่างๆ

Young C. Song และคณะ [7] ได้สร้างเวิร์คโฟลว์สำหรับข้อมูล Biological pathway (กลไกการทำงานระหว่างยีน) และ Non-synonymous SNPs ซึ่งใช้ข้อมูลจากเว็บเซอร์วิส BioMoby และใช้โปรแกรมทาเวอร์นาในการสร้างและออกแบบให้ได้มาเป็นเวิร์คโฟลว์ โปรแกรมที่ Young C. Song และคณะทำการสร้างขึ้นมีชื่อว่า โปรแกรม DataBiNS [7] หน้าที่หลักของโปรแกรม คือรวบรวมข้อมูล Non-synonymous SNPs จากฐานข้อมูล dbSNP [27] กับข้อมูลที่เกี่ยวข้องกับกลไกการทำงานระหว่างยีน (Biological pathway) ต่างๆจากฐานข้อมูล KEGG [28] จากนั้นจะดึงข้อมูลรายการผลงานที่ถูกตีพิมพ์, Gene ontology, Gene annotation และข้อมูล nsSNP [29] ของแต่ละยีนใน Biological pathway ต่างๆนั้น นอกจากนี้โปรแกรมยังนำเอาโปรแกรม LS-SNP [7][30] ที่มีความสามารถในการทำนายผลกระทบของการเปลี่ยนแปลงสนิปที่อาจส่งผลถึงการทำงานของโปรตีนมารวมไว้ด้วยกัน ข้อมูลทั้งหมดจะถูกดึงมาแสดงภายในการสั่งงานเพียงครั้งเดียว สิ่งที่ได้จากโปรแกรม คือ โปรแกรมจะช่วยลดหรือคัดกรองจำนวนและรวมทั้งตำแหน่งของสนิป ที่มีความเกี่ยวข้องกับกลไกการทำงานระหว่างยีนน้อยๆออก โดยเหลือเฉพาะตำแหน่งที่สามารถนำไปวิเคราะห์ต่อในระดับหน้าที่การทำงานได้

การเลือกตำแหน่งสนิปที่น่าสนใจจากผลที่ได้จากสนิปชิป รวมทั้งการหาความสัมพันธ์หรือการทำนายถึงผลกระทบ อันเนื่องมาจากการเปลี่ยนแปลงของสนิปในมนุษย์กับการตอบสนองต่อยาหรือภาวะการเป็นโรค เป็นสิ่งที่จำเป็นต้องทำหลังจากที่ได้ผลการวิเคราะห์ทางห้องปฏิบัติการมาแล้ว ข้อมูลที่มีมากมายในปัจจุบันจากหลายฐานข้อมูลที่เปิดให้ใช้งานได้ในขณะนี้สามารถทำให้เกิดองค์ความรู้ใหม่ๆในด้านการศึกษาสนิป อย่างไรก็ตามผลจากการค้นหาข้อมูลจากฐานข้อมูลในแต่ละฐานข้อมูลดังกล่าวไม่ได้อยู่ในรูปแบบที่สามารถใช้งานได้ง่าย หรือฐานข้อมูลเดียวอาจไม่เพียงพอในการสร้างเป็นองค์ความรู้ได้ คณะผู้วิจัยจึงมีวัตถุประสงค์ที่จะพัฒนาวิธีการทำนายผลกระทบอันเนื่องมาจากการสนิปต่างๆ รวมทั้งสร้างระบบที่มีความสามารถในการคัดเลือกตำแหน่งสนิปที่น่าสนใจในการนำไปศึกษาต่อทางด้านหน้าที่การทำงาน (Functional Study) ต่อไป ซึ่งยังไม่มีผู้ใดเคยทำมาก่อน คณะผู้จัยจะทำการศึกษาสนิปที่ละเอียดครอบคลุมและใช้งานง่ายโดยการแก้ปัญหางานวิจัยนี้จะใช้โปรแกรมทาเวอร์นา ที่มีความสามารถในการสร้างและจัดการรูปแบบข้อมูล รวมทั้งมีความสามารถในการประสานการทำงานกับเว็บเซอร์วิสต่างๆ และแสดงผลออกมาในรูปแบบที่ง่ายต่อการนำไปศึกษาหรือใช้งานต่อไป จึงเป็นเหตุให้เกิดความร่วมมือในการวิจัยระหว่างมหาวิทยาลัยและเป็นกรณีศึกษาของวิทยานิพนธ์นี้

1.3 วัตถุประสงค์

1.3.1. พัฒนาเวิร์คโฟลว์และสร้างการเชื่อมต่อในเวิร์คโฟลว์ เพื่อแก้ไขปัญหาในงานวิจัยการทำนายผลกระทบของสนิปในงานวิจัยทางด้านเภสัชพันธุศาสตร์ และการวิเคราะห์สนิปของกุ้ง โดยใช้เทคโนโลยีมายกริด, โปรแกรมทาเวอร์นา, เว็บเซอร์วิสและใช้เวิร์คโฟลว์ทั้งสองนี้เป็นกรณีศึกษาของงานวิทยานิพนธ์

1.3.2. นำเสนอกลไกในการลดเวลาในการพัฒนา และทดสอบเวิร์คโฟลว์โดยพัฒนาการตรวจสอบความถูกต้องของเวิร์คโฟลว์ก่อนการทำงานจริง ซึ่งจะมุ่งเน้นไปที่การตรวจสอบบริการไบโอมาร์ทภายใต้สิ่งแวดล้อมของโปรแกรมทาเวอร์นา ที่สามารถตรวจสอบคุณสมบัติของบริการไบโอมาร์ทได้ว่าทันสมัยพร้อมทำงานหรือล้าสมัยไปแล้ว ซึ่งสามารถระบุบริการที่ล้าสมัยไปแล้วในเวิร์คโฟลว์ และให้ข้อมูลที่มีนัยสำคัญต่อผู้ใช้งานจะได้มีความสะดวกและง่ายต่อการปรับแก้เวิร์คโฟลว์ซึ่งสามารถลดเวลาที่เสียไปได้อย่างมาก

1.3.3. นำเสนอกลไกในการลดปัญหาการเกินเวลา และความไม่คงเส้นคงวาในการทำงานที่ส่งผลให้เวิร์คโฟลว์ทำงานไม่สำเร็จโดยพัฒนาบริการท้องถิ่นขึ้นมาใช้งานเอง (Local web services)

1.4 ขอบเขตของงานวิจัย

1.4.1. พัฒนาเวิร์คโฟลว์โดยใช้โปรแกรทาเวอร์นารุ่น 1.5.0 – 1.7.1 และมีกรณีศึกษาคือเวิร์คโฟลว์ในการวิเคราะห์สนิปของกุ้งและของมนุษย์

1.4.2. พัฒนาการตรวจสอบเวิร์คโฟลว์ก่อนการทำงาน โดยจะมุ่งเน้นไปที่การตรวจสอบบริการไบโอมาร์ทภายใต้สิ่งแวดล้อมของโปรแกรมทาเวอร์นา โดยสามารถตรวจสอบคุณสมบัติของบริการไบโอมาร์ทได้ โดยมีข้อกำหนดในการตรวจสอบเป็น 1 บริการใช้ 1 ฐานข้อมูล

1.4.3. เพิ่มบริการท้องถิ่น (Local web services) เพื่อลดเวลาในการทำงานของเวิร์คโฟลว์ด้วยเทคโนโลยีมายกริดและเว็บเซอร์วิส

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1. ได้วิธีการแก้ไขปัญหาลดความล่าช้าในการทำงานของเว็บเซอร์วิส ด้านชีวสารสนเทศโดยการใช้บริการท้องถิ่นบนโปรแกรมทาเวอร์นา

1.6.2. ได้เวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท ภายใต้สิ่งแวดล้อมของทาเวอร์นาโดยสามารถตรวจสอบคุณสมบัติของบริการไบโอมาร์ทได้

1.6.3. ได้เวิร์คโฟลว์ในการแก้ปัญหาในงานวิจัย การทำนายผลกระทบของการเปลี่ยนแปลงลำดับเบส (SNP) ของมนุษย์ในงานวิจัยเภสัชพันธุศาสตร์

1.6.4. ได้องค์ความรู้เกี่ยวกับการพัฒนาเวิร์คโฟลว์ ด้านชีวสนเทศภายใต้สิ่งแวดล้อมของทาเวอร์นา

1.6 สรุป

วิทยานิพนธ์นี้เป็นการประยุกต์ใช้เทคโนโลยีมายกริด และเว็บเซอร์วิสเพื่อแก้ไขปัญหาในงานวิจัยทางด้านชีวสารสนเทศศาสตร์ ซึ่งเป็นงานวิจัยที่ทำร่วมกับหน่วยงานวิจัยที่มีความชำนาญทางด้านชีวสารสนเทศศาสตร์ของมหาวิทยาลัยสงขลานครินทร์และ โรงพยาบาลรามาธิบดี คณะแพทยศาสตร์ มหาวิทยาลัยมหิดล

สำหรับบทต่อไปจะกล่าวถึงภาพรวม และสถาปัตยกรรมของมายกริดในงานวิทยานิพนธ์ตลอดจนการอธิบายรายละเอียดของบริการไบโอมาร์ท บทที่ 3 จะกล่าวถึงการเลือกบริการที่เหมาะสม โดยใช้เวิร์คโฟลว์การสร้างโครงสร้างต้นไม้สายวิวัฒนาการของกุ้ง ซึ่งเป็นการทำวิจัยร่วมกับศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์ เป็นกรณีศึกษาแรก

บทที่ 4 จะกล่าวถึงการออกแบบและพัฒนาเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ ซึ่งเป็นการทำวิจัยร่วมกับศูนย์ความเป็นเลิศด้านชีววิทยาศาสตร์ของประเทศไทยหรือ Thailand Center of Excellence for Life Sciences (TCELS) โรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล การออกแบบและพัฒนาเวิร์คโฟลว์สำหรับตรวจสอบความถูกต้องของเวิร์คโฟลว์นี้จะกล่าวในบทที่ 5 ผลการทดลองจะกล่าวในบทที่ 6 และสรุปงานวิทยานิพนธ์ในบทที่ 7

บทที่ 2

สถาปัตยกรรม โปรแกรมทาเวอร์นาและเวิร์คโฟลว์

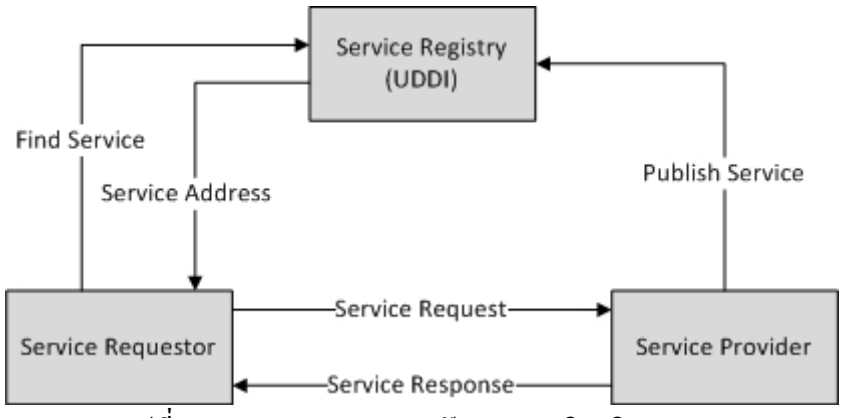
บทนี้จะกล่าวถึงคือสถาปัตยกรรมของงานวิจัย และหลักการสำคัญที่เกี่ยวข้องกับเวิร์คโฟลว์, โปรแกรมทาเวอร์นา, เว็บเซอร์วิสและบริการต่างๆที่จะนำมาสร้างเป็นเวิร์คโฟลว์ รวมทั้งรายละเอียดเกี่ยวกับบริการไอโบมาร์ท

2.1 สถาปัตยกรรมที่เกี่ยวข้องกับวิทยานิพนธ์

งานวิจัยในวิทยานิพนธ์นี้ได้ประยุกต์ใช้สถาปัตยกรรมต่างๆดังนี้

2.1.1 การพัฒนาระบบบนเครือข่ายกริดด้วยแนวความคิดของสถาปัตยกรรมเชิงบริการ

การพัฒนาระบบซอฟต์แวร์เพื่อใช้งานกันระหว่างองค์กรบนเครือข่ายกริด มีปัจจัยที่ต้องนำมาใช้ประกอบการพิจารณาอยู่มากมายด้วยกัน และปัจจัยหลักที่ต้องพิจารณาเป็นอันดับต้นๆก็คือ วิธีการออกแบบระบบที่สามารถเข้ากันได้กับเทคโนโลยีของทุกๆองค์กรที่ร่วมมือกัน รวมไปถึงต้องสามารถเข้ากันได้กับวัฒนธรรมขององค์กรนั้นๆด้วย โดยคำว่าวัฒนธรรมขององค์กรนั้นเกี่ยวข้องกับวิธีการและขั้นตอนในการดำเนินงานของแต่ละองค์กรมีความแตกต่างกัน และระบบที่ใช้อยู่เดิมของแต่ละองค์กรนั้น (Legacy system) มีความแตกต่างกันโดยสิ้นเชิงเป็นต้น ซึ่งปัจจัยทางด้านวัฒนธรรมนี้เอง ได้ผลักดันให้เกิดรูปแบบการพัฒนาระบบที่ตั้งอยู่บนฐานของสถาปัตยกรรมเชิงการบริการหรือ Service Oriented Architecture (SOA) [31] โดยองค์ประกอบของ SOA มี 3 ส่วนคือผู้ใช้บริการ (Service Requestor) ผู้ให้บริการ (Service Provider) และฝ่ายทะเบียนของบริการ (Service Registry) [31] โดยผู้ให้บริการจะเป็นผู้จัดสรรบริการให้แก่ผู้ใช้บริการ และฝ่ายทะเบียนของบริการเปรียบเสมือนสมุดหน้าเหลืองสำหรับค้นหาบริการที่จัดสรรโดยผู้ให้บริการ ซึ่งผู้ใช้บริการสามารถค้นหาบริการที่ต้องการได้จากหน่วยนี้ดังรูปที่ 2.1



รูปที่ 2.1 การทำงานของสถาปัตยกรรมเชิงบริการ [30]

สถาปัตยกรรมเชิงบริการ ประกอบด้วยกลุ่มของบริการที่ติดต่อสื่อสารระหว่างกันได้ การติดต่อสื่อสารนี้สามารถส่งผ่านข้อมูลระหว่างบริการสองบริการหรือมากกว่า เพื่อประสานงานกันในการทำงานต่างๆ เริ่มแรกนั้นสถาปัตยกรรมเชิงบริการคือการใช้ Distributed Component Object Model (DCOM) หรือ Object Request Broker (ORBs) ที่อยู่บนพื้นฐานข้อกำหนดของ Common Object Request Broker Architecture (CORBA) [32]

2.1.2 เว็บเซอร์วิส

เว็บเซอร์วิส (Web Service) [32] คือโปรแกรมประยุกต์ที่สร้างให้รองรับการเรียกใช้งานจากโปรแกรมประยุกต์อื่นๆบนอินเตอร์เน็ต โดยสื่อสารกันด้วยข้อมูลที่อยู่ในรูปแบบ Extensible Markup Language (XML) รูปแบบ XML ที่ใช้นี้กำหนดเป็นมาตรฐานชื่อว่า Simple Object Access Protocol (SOAP) [33] โดยข้อมูลอาจถูกส่งผ่านทางโปรโตคอล Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) หรือ File Transfer Protocol (FTP) แต่ที่นิยมใช้มาก คือ HTTP เว็บเซอร์วิสประกอบด้วยดังนี้

- แอ็พพลิเคชัน คือ โปรแกรมประยุกต์ที่ทำหน้าที่ให้บริการอยู่บนเครื่องแม่ข่ายที่เปิดให้บริการตลอดเวลา สามารถติดต่อด้วยโพรโตคอล HTTP และพัฒนาด้วยภาษาที่มีความสามารถในการจัดการกับ SOAP ได้
- SOAP คือ โปรโตคอลหรือระเบียบวิธีในการติดต่อสื่อสารระหว่างเว็บเซอร์วิส โดยใช้ข้อมูลที่กำหนดรูปแบบด้วยภาษา XML ทำให้เว็บเซอร์วิส

สามารถสื่อสารกันได้ แม้ว่าจะอยู่บนเครื่องคอมพิวเตอร์คนละแพล็ตฟอร์ม หรือพัฒนาด้วยภาษาโปรแกรมที่แตกต่างกัน เมื่อแอ็พพลิเคชันต้องการใช้งานเว็บเซอร์วิส ต้องเขียนโปรแกรมเพื่อติดต่อกับ SOAP ในภาษาที่ตนใช้ จากนั้น SOAP ก็จะสร้าง SOAP message เพื่อติดต่อกับแอ็พพลิเคชันปลายทางให้โดยอัตโนมัติ

- Web Services Description Language (WSDL) [34] คือ เอกสาร XML ที่อธิบายรายละเอียดในการติดต่อกับเว็บเซอร์วิส เพื่อให้แอ็พพลิเคชันที่ต้องการเรียกใช้เว็บเซอร์วิสรู้ว่า เซอร์วิสนั้นให้บริการอะไรบ้าง และจะติดต่อได้อย่างไร
- UDDI (Universal Description, Discovery and Integration) คือ ไดเรกทอรีที่เก็บรวบรวมเว็บเซอร์วิสที่มีการลงทะเบียนไว้ UDDI จะเก็บรวบรวมข้อมูลของเว็บเซอร์วิสต่างๆไว้ในรูปแบบ WSDL หน้าที่ของ UDDI จะคล้ายกับเว็บไดเรกทอรีหรือเว็บสารบัญ กล่าวคือ UDDI ช่วยให้ผู้พัฒนาเว็บเซอร์วิสได้ประกาศหรือประชาสัมพันธ์บริการของตนเองสู่สาธารณะ และช่วยให้ผู้ใช้งานเว็บเซอร์วิสค้นพบเว็บเซอร์วิสที่ต้องการใช้งานได้
  - รูปที่ 2.2 อธิบายการทำงานของเว็บเซอร์วิสได้ดังนี้

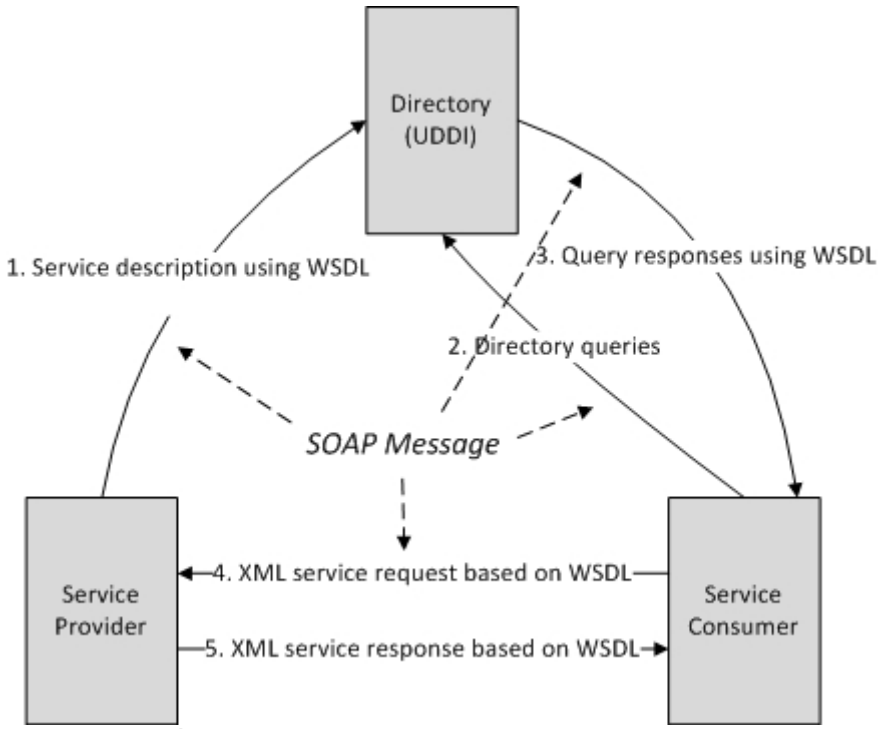
    1) ผู้ให้บริการจะให้คำอธิบายการใช้บริการโดยใช้ WSDL ไว้ที่ไดเรคทอรีของเว็บเซอร์วิสหรือ UDDI

    2) ผู้ขอใช้บริการจะค้นหาที่ต้องการจาก UDDI ทำให้ทราบว่าบริการอยู่ที่ไหน และจะติดต่อกับบริการนั้นได้อย่างไร

    3) UDDI จะตอบกลับด้วยเอกสาร WSDL กลับมาให้ผู้ขอใช้บริการ โดยจะบอกว่าบริการที่จะเรียกใช้ต้องร้องขอ และจะได้ผลลัพธ์อย่างไร

    4) ผู้ขอใช้บริการใช้ WSDL ส่งการร้องขอการใช้งานไปยังผู้ให้บริการ

    5) ผู้ให้บริการตอบกลับตามที่ผู้ขอใช้บริการร้องขอ



รูปที่ 2.2 สถาปัตยกรรมการทำงานของเว็บเซอร์วิส [32]

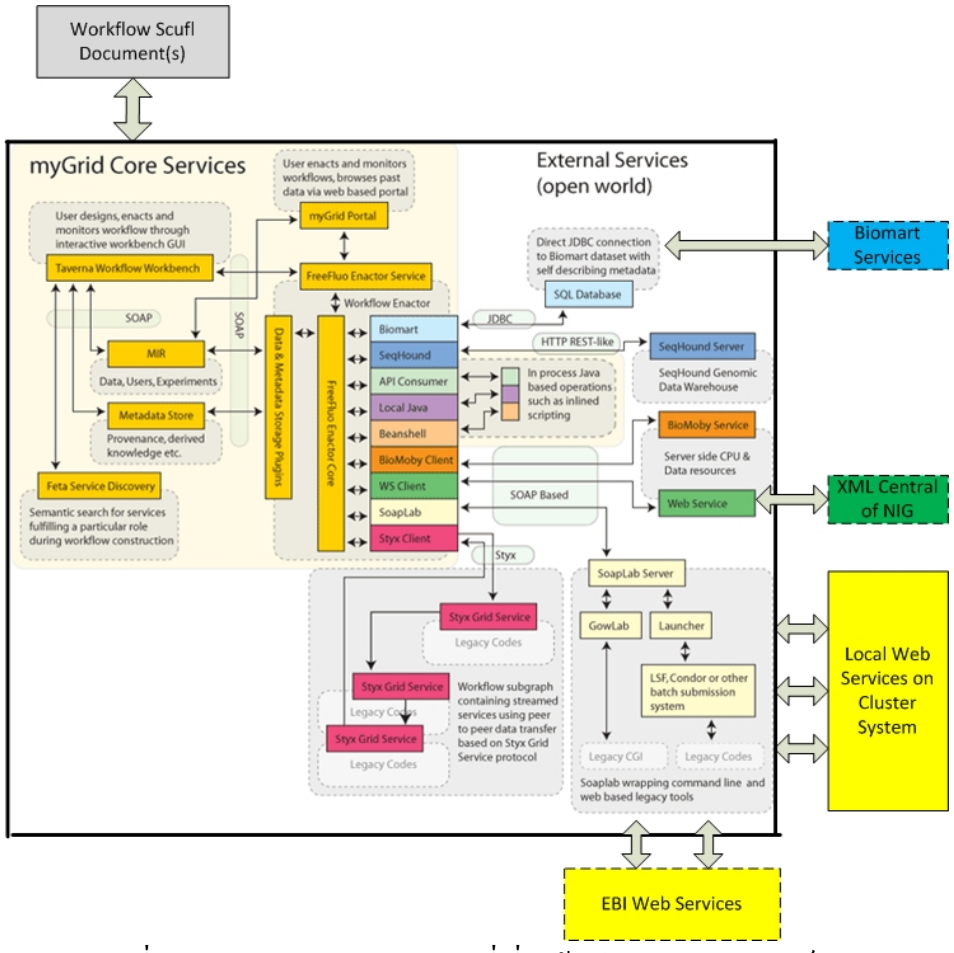
2.1.3 สถาปัตยกรรมของโปรแกรมทาเวอร์นา

รูปที่ 2.3 เป็นสรุปการติดต่อสื่อสารและเชื่อมโยงของคอมโพเนนท์อันหลากหลายของมายกริดและทาเวอร์นา บริการที่นำเสนอโดยโครงการมายกริดแสดงด้วยบล็อกสีเหลืองเข้ม ส่วนบริการอื่นๆซึ่งอยู่กระจัดกระจายบนเครือข่ายอินเตอร์เน็ต เป็นบริการที่โครงการมายกริดมีข้อตกลงร่วมกันในการสื่อสารซึ่งกันและกันไว้ บริการต่างๆเหล่านี้สามารถเข้าถึงได้โดยสาธารณะด้วยเครื่องมือหรือการโปรแกรมใดๆก็ได้ และเป็นบริการที่ได้รับการเชื่อมประสานมากับโปรแกรมทาเวอร์นาแล้ว ซึ่งอาจมองว่าบริการภายนอกเหล่านี้เป็นปลักอินหรือตัวเสริมโปรแกรมในโปรแกรมทาเวอร์นา

ในงานวิทยานิพนธ์นี้ได้สร้างบริการหรือเว็บเซอร์วิสขึ้นมาใช้งานเอง ในการทำนายโครงสร้างต้นไม้สายวิวัฒนาการของกุ้ง ซึ่งเป็นงานวิจัยร่วมกับศูนย์วิจัยจีโนมิกส์และชีว-สารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยทำงานอยู่บนระบบคลัสเตอร์ของศูนย์กริดแห่งมหาวิทยาลัยสงขลานครินทร์ ระบบคลัสเตอร์ติดตั้งระบบ Rocks Cluster รุ่น 4.3 [35] และสร้างส่วนประสานการเชื่อมต่อของเว็บเซอร์วิส ที่สร้างขึ้นมาใช้งานเองกับโปรแกรมทาเวอร์

นาโดยใช้โปรแกรม Soablab Analysis Tool [37] ซึ่งแสดงด้วยบล็อกสีเหลืองในไดอะแกรม นอกจากนี้ยังเลือกใช้บริการอื่นๆที่จำเป็น และมีการทำงานที่ถูกต้องตรงตามความต้องการในการทำนายโครงสร้างต้นไม้สายวิวัฒนาการของกุ้งจากบริการของสถาบันชีวสารสนเทศแห่งยุโรป หรือ European Bioinformatics Institute (EBI) [39] ด้วย

เวิร์คโฟลว์ต่างๆในรูปแบบของเอกสาร Simple Conceptual Unified Flow Language (Scufl) [9] ที่สร้างในงานวิทยานิพนธ์แสดงด้วยบล็อกสีน้ำเงิน ซึ่งเมื่อพิจารณาในเชิงสถาปัตยกรรมแล้ว เวิร์คโฟลว์ก็เป็นส่วนหนึ่งของการทำงานภายนอกหรือ Open world ก็ได้ ในวิทยานิพนธ์นี้ได้เรียกใช้บริการชนิดหลักๆ คือ เว็บเซอร์วิสมาตรฐานแสดงด้วยบล็อกสีเขียว ตัวอย่างเช่น Web API for Biology (WABI) ของ DNA Data Bank of Japan (DDBJ) [40] และบริการไบโอมาร์ทแสดงด้วยบล็อกสีฟ้า โดยเวิร์คโฟลว์เรียกใช้งานบริการต่างๆผ่านแกนบริการกลางของมายกริด (myGrid Core Services) ไปยังเว็บเซอร์วิสหรือบริการต่างๆที่กระจัดกระจายกันอยู่บนอินเตอร์เน็ต



รูปที่ 2.3 สถาปัตยกรรมของระบบที่เกี่ยวข้องในงานวิทยานิพนธ์ [9]

2.2 โปรแกรมทาเวอร์นาและเวิร์คโฟลว์

ต่อไปจะกล่าวถึงภาพรวมและส่วนประกอบของโปรแกรมทาเวอร์นาดังนี้

2.2.1 ภาพรวมของโปรแกรมทาเวอร์นา

โปรแกรมทาเวอร์นาอำนวยความสะดวกให้ผู้ใช้งาน สามารถสร้างเวิร์คโฟลว์ในการวิเคราะห์ระบบงานที่ซับซ้อนได้จากคอมโพเนนท์หรือบริการต่างๆทั้งที่อยู่ในท้องถิ่น หรือกระจัดกระจายอยู่บนอินเตอร์เน็ต แล้วสั่งให้ประมวลผลเพื่อให้ได้ผลลัพธ์ในรูปแบบที่เข้าใจได้ง่าย ในการสนับสนุนกลไกหน้าที่การทำงานเหล่านี้ โปรแกรมทาเวอร์นาจึงมีคอมโพเนนท์ต่างๆมารองรับการทำงาน อย่างเช่นการค้นหา, การให้คำอธิบาย, และการเลือกบริการต่างๆที่เกี่ยวข้องกับการทำงานตามความต้องการนั้นๆ [2]

1) คำนิยาม

คำนิยามของคำศัพท์ที่ใช้ในสภาพแวดล้อมของทาเวอร์นาในวิทยานิพนธ์มีดังนี้

- **เวิร์คโฟลว์ (Workflow)** – เป็นกลุ่มของคอมโพเนนท์และความสัมพันธ์ต่างๆ ที่กำหนดหรือบอกถึงกระบวนการที่ซับซ้อนกระบวนหนึ่งจากการแสดงผลด้วยบล็อกอย่างง่าย ความสัมพันธ์ต่างๆนี้อาจจะอยู่ในรูปแบบของการเชื่อมโยงของข้อมูล (Data links), การส่งต่อข้อมูลซึ่งเป็นเอาท์พุตของคอมโพเนนท์หนึ่งๆ ไปเป็นอินพุตให้กับคอมโพเนนท์อื่นๆ (Data flow) หรืออยู่ในรูปแบบของการควบคุมการเชื่อมต่อ ที่บอกเงื่อนไขการทำงานของคอมโพเนนท์นั้นๆ (Control links) ตัวอย่างเช่น คอมโพเนนท์เอจะทำงานก็ต่อเมื่อคอมโพเนนท์บีทำงานเสร็จแล้ว ในสิ่งแวดล้อมของทาเวอร์นา เวิร์คโฟลว์หนึ่งๆแสดงรูปแบบการไหลของข้อมูลหรือ Workflow data model ซึ่งอยู่ที่เครื่องส่วนบุคคลหรือบนเว็บไซต์ก็ได้โดยเป็นไฟล์ XML ในรูปแบบของเอกสาร Simple Conceptual Unified Flow Language (Scufl)
- **คอมโพเนนท์ (Component)** – คอมโพเนนท์จะใช้ในการสร้างบล็อกที่ทำหน้าที่อย่างใดอย่างหนึ่งซึ่งกำหนดไว้แล้วภายในกระบวนการนั้นๆ เราสามารถมองได้ว่าเครื่องมือรับคำสั่ง (Command line tool) ใดๆ หรือสคริปต์ใดๆ ของภาษา Perl คือหนึ่งคอมโพเนนท์ คอมโพเนนท์เหล่านี้ควรมีลักษณะการทำงานแบบอะตอมมิกหรือปรมาณู (Atomic) อธิบายลักษณะเช่น

ไม่สามารถแบ่งย่อยได้มากกว่านี้อีกแล้ว คอมโพเนนท์อาจจะเป็นหน่วยที่ทั้งใช้ข้อมูลและผลิตข้อมูลก็ได้ ตัวอย่างเช่น งานของโปรแกรม Basic Local Alignment Search Tool (BLAST) [41] หนึ่งๆ คือหนึ่งคอมโพเนนท์ที่ใช้ข้อมูลลำดับเบสและพารามิเตอร์สำหรับการค้นหาอื่นๆ และผลลัพธ์ก็คือผลิตรายงานที่บอกว่าลำดับเบสใดพบในฐานข้อมูลใดบ้าง และมีรายละเอียดอย่างไร เป็นต้น คอมโพเนนท์เหล่านี้อาจติดตั้งอยู่บนระบบที่มีพลังการคำนวณในระดับสูง หรือแม้แต่ในคอมพิวเตอร์ส่วนบุคคลของผู้ใช้งานเอง โดยสามารถเข้าถึงได้ง่ายผ่านเครือข่ายอินเตอร์เน็ต

- **บริการ (Service)** – บริการอีกชื่อหนึ่งก็คือคอมโพเนนท์ แต่ใช้คำว่าบริการในกรอบอ้างอิงว่า บริการเหล่านั้นทำงานอยู่บนเครื่องบริการหรือเซอร์เวอร์ที่มีกำลังการประมวลผลสูงซึ่งไม่ได้เป็นเครื่องในท้องถิ่นของผู้ใช้เอง หรือบางกรณีที่กล่าวถึงบริการที่ปรากฏในเวิร์คโฟลว์ของโปรแกรมทาเวอร์นา อาจจะใช้คำว่า โปรเซสเซอร์ (Processor) ก็ได้

- **เอ็นเนกเตอร์ (Enactor)** – เอ็นเนกเตอร์ของเวิร์คโฟลว์คือหน่วยที่ทำหน้าที่ในการประสานงานการทำงานของคอมโพเนนท์ต่างๆในเวิร์คโฟลว์ โดยอาจจะเรียกได้ว่าเป็นบริการๆหนึ่งโดยตัวมันเองในกรณีที่มันใช้เอกสารเวิร์คโฟลว์ และพารามิเตอร์ต่างๆเป็น อินพุต และผลิตผลลัพธ์ออกมา หรือในกรณีของทาเวอร์นาอาจเป็นคอมโพเนนท์หนึ่งภายในชุดของโปรแกรม เอ็นเนกเตอร์จะจัดการกระบวนการในภาพรวมคือการรายงานสถานะการทำงาน, การส่งผ่านข้อมูลระหว่างคอมโพเนนท์และงานอื่นๆที่เกี่ยวข้องทั้งหมด

2) ประโยชน์ของการใช้เทคโนโลยีเวิร์คโฟลว์

โปรแกรมทาเวอร์นาสร้างขึ้นด้วยเทคโนโลยีใหม่ทั้งหมดเช่น Web service และ Ontology เป็นต้น โดยเกิดจากการพยายามลดอุปสรรคในการใช้งานของผู้ใช้ที่ต้องเสียเวลาไปมากกับการทำงานวิจัยบนฐานข้อมูลที่ซับซ้อนที่กระจายอยู่บนอินเตอร์เน็ต ประโยชน์ของการใช้เทคโนโลยีเวิร์คโฟลว์ มีดังนี้ [8][9]

- **เพิ่มประสิทธิผล (Efficiency)** – โปรแกรมทาเวอร์นาช่วยประหยัดเวลาของผู้ใช้งานในหลายๆกรณีด้วยเทคโนโลยีทางด้านเวิร์คโฟลว์

- **ช่วยในการออกแบบการวิเคราะห์ (Analysis Design)** – การออกแบบการวิเคราะห์แบบเวิร์คโฟลว์จะสามารถทำได้รวดเร็วกว่าวิธีอื่นๆที่ผ่านมา ทำให้มองเห็นการวิเคราะห์ในภาพรวมได้ และยังสามารถเพิ่มคอมโพเนนท์ใหม่ๆเพื่อเป็นการขยายการวิเคราะห์ได้โดยง่าย

- **การเรียกใช้การทดลองแบบ *in silico* จากระยะไกล (Experiment Invocation)** – การทำงานงานด้านชีวสารสนเทศแบบดั้งเดิมส่วนมาก ตั้งแต่ขนาดเล็กรวมไปถึงโครงการการทำเครื่องหมายจีโนม (Whole genome annotation) ทำโดยการผสมผสานกันระหว่างเว็บเบราเซอร์และการใช้งานคำสั่งบนระบบปฏิบัติการ UNIX หรือเครื่องมือการทำงานอื่นๆ การผสมผสานของเครื่องมือเหล่านี้ ต้องการส่งผ่านข้อมูลระหว่างคอมโพเนนท์ต่างๆโดยการคัดลอกแล้ววาง ลักษณะการทำงานแบบนี้ใช้เวลามาก มีแนวโน้มที่จะเกิดข้อผิดพลาดได้ง่าย และไม่เป็นระบบการทำงานที่ดี โปรแกรมทาเวอร์นามีความสามารถในการรวบรวมคอมโพเนนท์ต่างๆเหล่านี้เข้าด้วยกันเป็นเวิร์คโฟลว์และเรียกสั่งงานจากระยะไกลได้
- **จัดการคอมโพเนนท์ (Component Management)** – ในยุคปัจจุบันมีความจำเป็นต้องเรียกใช้งานบริการฐานข้อมูลจากแหล่งระยะไกล เช่น สถาบันชีวสารสนเทศแห่งยุโรป หรือ European Bioinformatics Institute (EBI) ทำให้ผู้ใช้สามารถใช้งานบริการที่ทันสมัยอยู่เสมอๆ มีโปรแกรมหรือคอมโพเนนท์ต่างๆให้เลือกใช้ได้มากมาย และที่สำคัญคอมโพเนนท์เหล่านั้นทำงานบนระบบคลัสเตอร์ โปรแกรมทาเวอร์นามีความสามารถที่เข้าถึงเครื่องซุปเปอร์คอมพิวเตอร์เหล่านี้ได้โดยง่ายผ่านทางเครื่องคอมพิวเตอร์ส่วนบุคคลสมัยใหม่ เนื่องจากโครงการมายกริดมีความร่วมมือกับแหล่งฐานข้อมูลมาตรฐานระดับโลกต่างๆ

  การใช้งานเทคโนโลยีเวิร์คโฟลว์ ทำให้ไม่ต้องพัฒนาระบบงานที่ไม่จำเป็น เช่น อัลกอริทึมการทำนายโครงสร้างของโปรตีนขั้นที่สอง ก็จะต้องพัฒนาบริการที่เกี่ยวข้องทั้งหมดด้วยตัวเองเพื่อให้เพียงพอที่จะสามารถทำงานได้ ต้องสืบค้นข้อมูลลำดับเบสก่อน ซึ่งต้องมีงานด้านการจัดการฐานข้อมูลเข้ามาเกี่ยวข้องโดยเป็นการเพิ่มภาระงานโดยไม่จำเป็น แต่หากใช้เทคโนโลยีเวิร์คโฟลว์ ผู้ใช้สามารถเลือกบริการที่พร้อมทำงานโดยผู้ใช้ไม่จำเป็นต้องสนใจเรื่องการเข้าถึงฐานข้อมูลเลย ผู้ใช้งานสามารถเลือกแหล่งบริการที่เหมาะสมตามความต้องการซึ่งมีให้เลือกมากมาย เช่นผู้ให้บริการหลักทางด้านชีวสารสนเทศอย่างสถาบันชีวสารสนเทศแห่งยุโรป (EBI) หรือไบโอมาร์ท (BioMart)
- **การเข้าถึงระบบประมวลที่มีประสิทธิภาพสูง (Invocation Performance)** – แม้ว่าเครื่องคอมพิวเตอร์บุคคลสมัยใหม่จะมีฮาร์ดแวร์ที่ดีขึ้นตามลำดับ แต่อย่างไรก็ตามก็ยังต้องการอัลกอริทึมในการทำงานเพื่อดึงเอาพลังการประมวลผลจากฮาร์ดแวร์เหล่านั้นมาใช้ การใช้งานคอมโพเนนท์ในระยะไกล สามารถทำให้ผู้ใช้งานสามารถใช้ประโยชน์จากฮาร์ดแวร์ที่ทรงประสิทธิภาพ ตัวอย่างเช่นโปรแกรม InterProScan [42] ที่พัฒนาโดย EBI ไม่ได้ทำงานอยู่บนเครื่องคอมพิวเตอร์ธรรมดา แต่ทำงานบนเครื่องคลัสเตอร์ขนาดหลายร้อยโหนด ซึ่งเวิร์คโฟลว์ที่มีบริการ

ที่มีโครงสร้างพื้นฐานที่สนับสนุนการใช้บริการ จากแหล่งที่มีประสิทธิภาพการประมวลผลสูงแบบนี้ สามารถให้ผลลัพธ์แก่นักวิจัยที่รวดเร็วกว่าการใช้เครื่องคอมพิวเตอร์ธรรมดาอย่างมาก

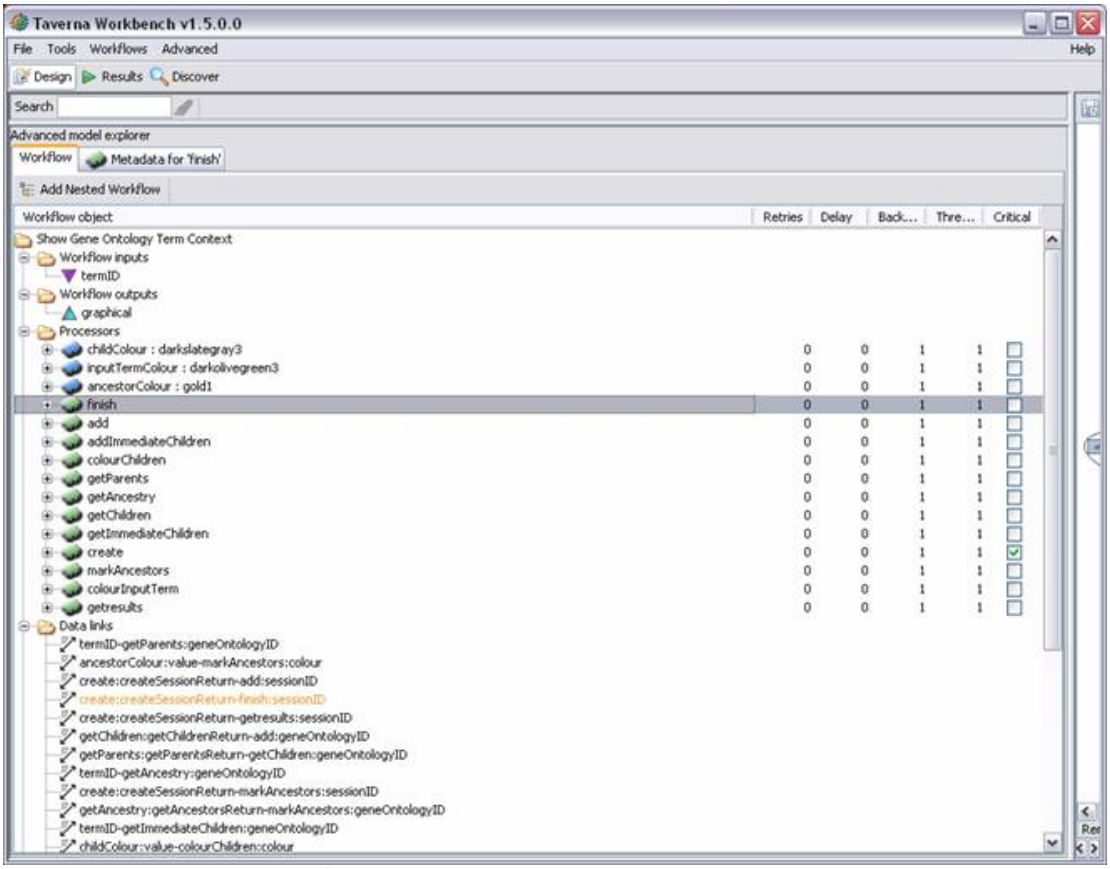
2.2.2 ส่วนประกอบของโปรแกรมทาเวอร์นา

ในส่วนนี้จะนำเสนอส่วนประกอบทางด้านกราฟิกต่างๆ ของโปรแกรมทาเวอร์นา และอธิบายการใช้งานเครื่องมือต่างๆในขั้นต้น เพื่อให้นำไปสู่ความเข้าใจในการออกแบบเวิร์คโฟลว์สำหรับแก้ปัญหาใดๆที่สนใจในวิทยานิพนธ์ ส่วนประกอบหลักๆของโปรแกรมทาเวอร์นามีดังนี้

1) Advanced Model Explorer (AME)

- **Entity Table**

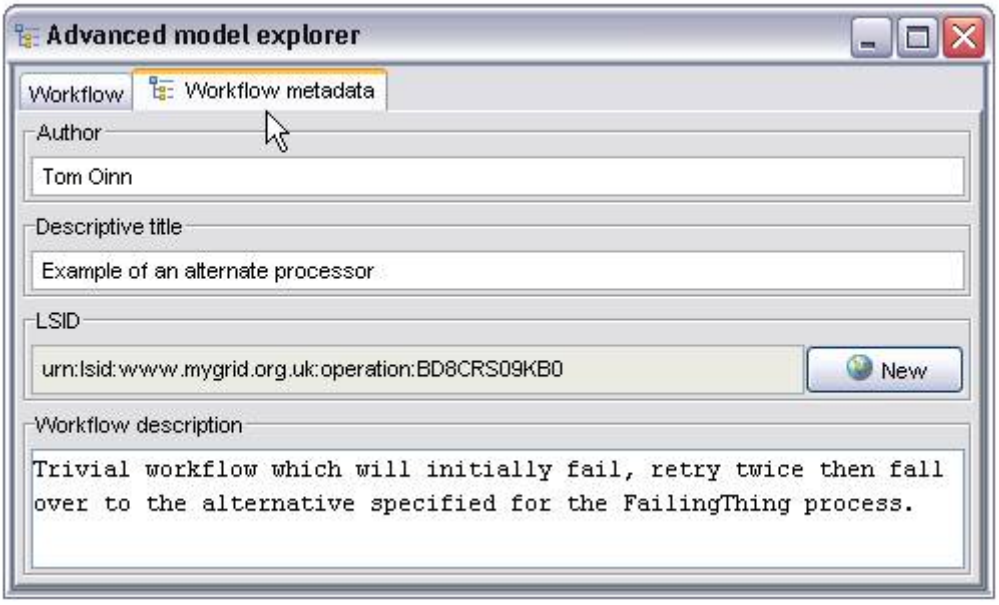
Entity Table เป็นส่วนหลักของ Advanced Model Explorer หรือ AME ซึ่งแสดงโปรเซสเซอร์หรือบริการ, อินพุตและเอาท์พุตของเวิร์คโฟลว์, การเชื่อมต่อของข้อมูล (Data flow link) และการประสานการเชื่อมต่อทั้งหมด โดยส่วนประกอบดังกล่าวนี้ อยู่ในรูปแบบโครงสร้างต้นไม้ ผู้ใช้สามารถแตกโครงสร้างต้นไม้เพื่อดูรายละเอียดได้ ดังรูปที่ 2.4



รูปที่ 2.4 หน้าต่าง AME ของโปรแกรมทาเวอร์นา

- **Workflow Metadata**

Workflow Metadata เป็นคำอธิบายการทำงานและจุดประสงค์ของเวิร์คโฟลว์ ทำให้มีการใช้เวิร์คโฟลว์ร่วมกันได้ คำอธิบายเวิร์คโฟลว์สามารถเพิ่มเติมหรือแก้ไขปรับปรุงได้โดยง่าย ตัวอย่างดังรูปที่ 2.5



รูปที่ 2.5 รายละเอียดของคำอธิบายใน AME

- **Workflow Inputs and Outputs**

เราสามารถสร้างพอร์ตของอินพุตและพอร์ตเอาท์พุต ได้ตามความต้องการที่จะรองรับข้อมูลจากบริการ ตัวอย่างเช่น บริการไบโอมาร์ทบางบริการให้เอาท์พุตหรือแอ็ตตริบิวต์จำนวนมาก ทั้งนี้สามารถที่จะเลือกได้ว่า จะเอาข้อมูลเหล่านี้มาใช้ต่อหรือไม่

นอกจากนี้ยังสามารถกำหนดชนิดของข้อมูล (Data Type) ให้กับอินพุตหรือเอาท์พุตของเวิร์คโฟลว์ได้ ซึ่งในโปรแกรมทาเวอร์นาจะเรียกว่า เมตาดาตา (Metadata) โดยสามารถที่จะกำหนดเพิ่มเติมได้ ซึ่งโปรแกรมทาเวอร์นาสามารถรองรับชนิดข้อมูลที่น่าสนใจดังตารางที่ 2.1

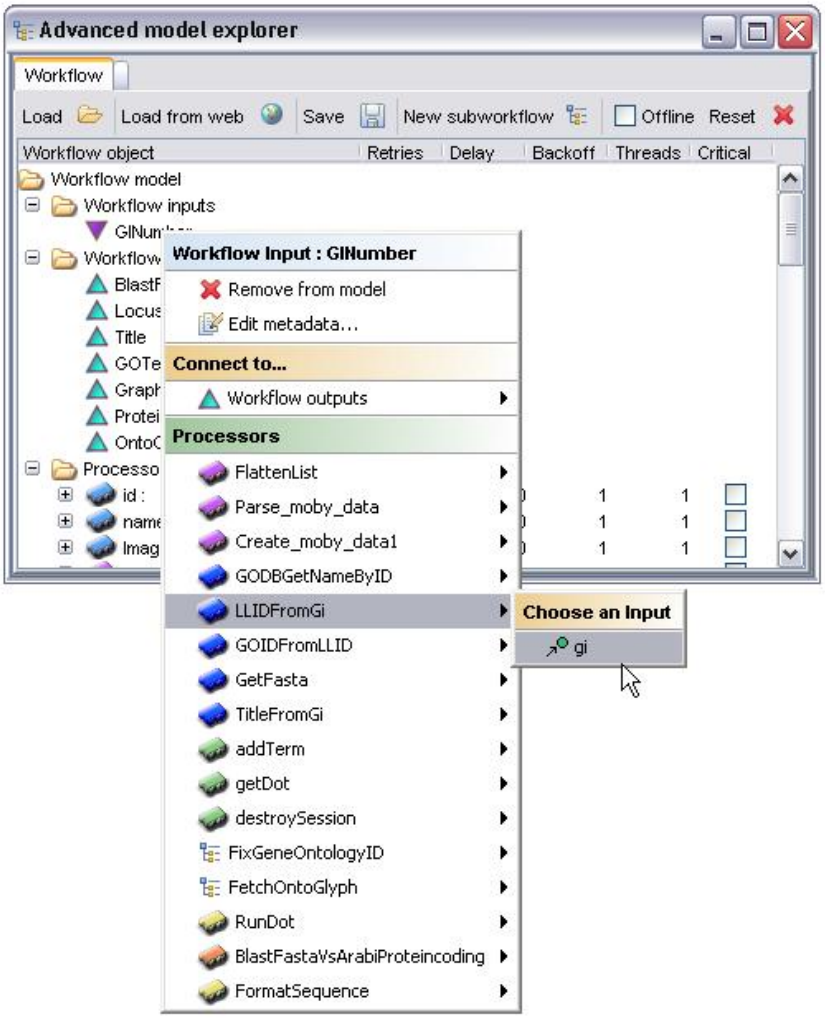
ตัวอย่างเช่น โปรแกรมทางด้านชีวสารสนเทศบางโปรแกรมให้เอาท์พุตที่เป็นกราฟิกไฟล์ชนิด PNG ดังนั้น สามารถกำหนด MIME Type เป็น image/png=PNG Image ให้กับเอาท์พุตนั้นได้ เป็นต้น

ตารางที่ 2.1 ชนิดข้อมูลที่โปรแกรมทาเวอร์นารองรับ

การกำหนดชนิดข้อมูล	ความหมาย
text/plain=Plain Text	ข้อความอักษรธรรมดา
text/xml=XML Text	ข้อความ XML
text/html=HTML Text	ข้อความ HTML
text/rtf=Rich Text Format	ข้อความอักษรธรรมดาพร้อมรูปแบบ
text/x-graphviz=Graphviz Dot File	ไฟล์ Dot ในระบบ Linux
image/png=PNG Image	ไฟล์ภาพสกุล PNG
image/jpeg=JPEG Image	ไฟล์ภาพสกุล JPEG
image/gif=GIF Image	ไฟล์ภาพสกุล GIF
application/zip=Zip File	ไฟล์บีบอัดสกุล ZIP
chemical/seq-aa-genpept=Genpept Protein	ข้อมูลลำดับของโปรตีน
chemical/seq-na-genbank=Genbank Nucleotide	ข้อมูลลำดับของนิวคลีโอไทด์
chemical/x-pdb=Protein Data Bank Flat File	ข้อมูลในรูปแบบโครงสร้างของโปรตีน

- **การเชื่อมต่ออินพุตของเวิร์คโฟลว์เข้ากับโปรเซสเซอร์**

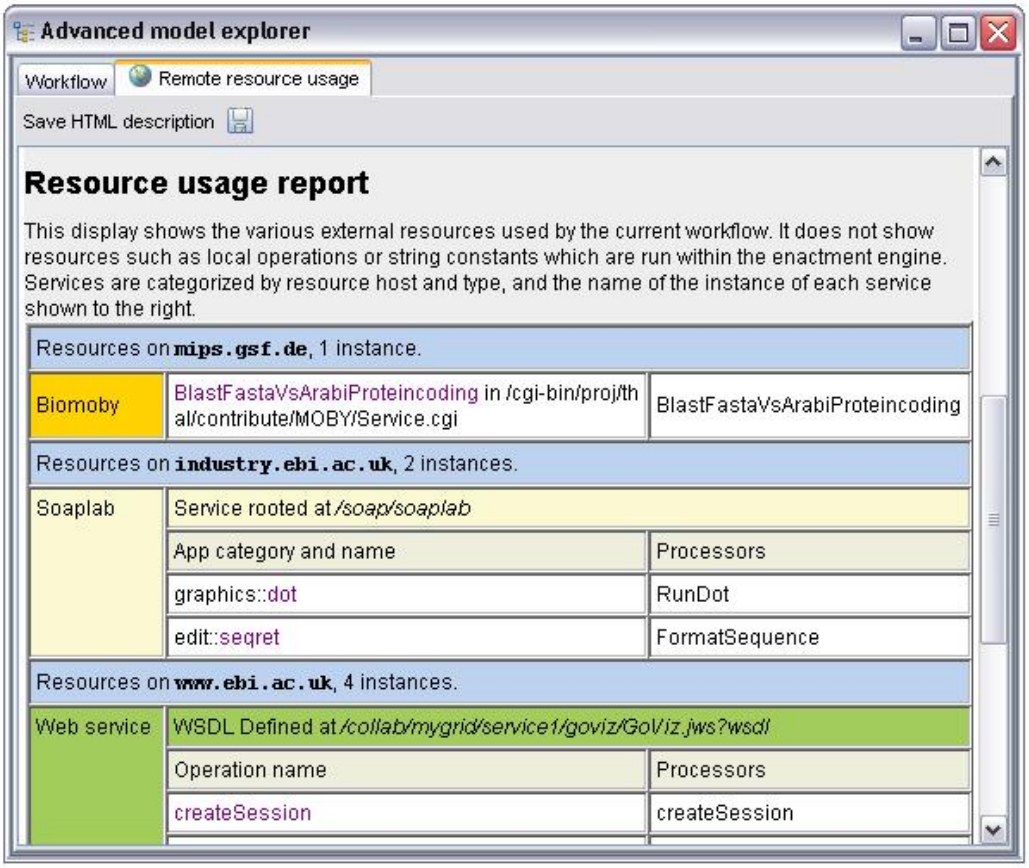
โปรแกรมทาเวอร์นาจะแสดงพอร์ตของอินพุตของโปรเซสเซอร์ใดๆ ทุกพอร์ตสามารถระบุได้ว่า เอาท์พุตนี้เข้าไปเป็นพอร์ตอินพุตใดของโปรเซสเซอร์ใด เพราะโปรเซสเซอร์หรือบริการต่างๆ นั้นไม่ได้รับอินพุตหรือพารามิเตอร์เพียงตัวเดียว หากแต่สามารถรับอินพุตหรือพารามิเตอร์หลายๆ ตัวในการทำงานซึ่งขึ้นอยู่กับการใช้งานและการออกแบบ ดังรูปที่ 2.6



รูปที่ 2.6 การเชื่อมต่ออินพุตเข้ากับโปรเซสเซอร์

- **โปรเซสเซอร์ (Processor)**

โปรเซสเซอร์เป็นโหนดระดับบนสุดในโปรแกรมทาเวอร์นา และเป็นคอมโพเนนท์ที่สำคัญที่สุด เพราะเป็นส่วนการทำงานของเวิร์คโฟลว์ ผู้ใช้งานสามารถดูรายละเอียดต่างๆ ของโปรเซสเซอร์ได้ ดังรูปที่ 2.7 ซึ่งแสดงผลสรุปให้ทราบว่าแต่ละโปรเซสเซอร์ว่าอยู่ที่ไหน (endpoint) เป็นบริการชนิดอะไร มีฟังก์ชันการทำงานอะไรบ้าง



รูปที่ 2.7 รายละเอียด Remote resource usage ของโปรเซสเซอร์ทั้งหมดที่ใช้ในเวิร์คโฟลว์

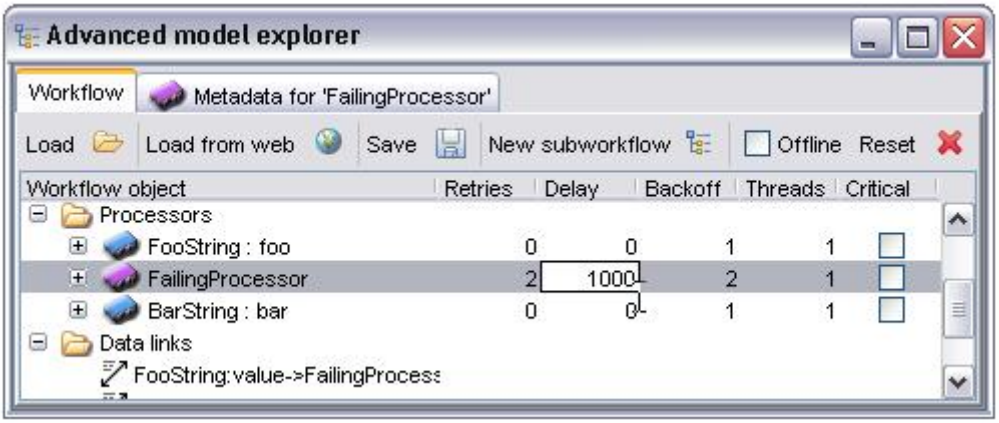
- **การกำหนดค่าการคงทนต่อการทำงาน**

การกำหนดค่าการคงทนต่อการทำงาน (Configuring Basic Fault Tolerance) ทำได้โดยการปรับแต่งค่าคุณสมบัติของแต่ละโปรเซสเซอร์ดังรูปที่ 2.8 อธิบายเฉพาะส่วนที่สำคัญดังนี้

- การทดลองใหม่ (Retries) การกำหนดจำนวนครั้งของทำงานซ้ำเมื่อการทำงานของโปรเซสเซอร์ล้มเหลว
- การถ่วงเวลา (Delay) ระหว่างการทดลองใหม่ในแต่ละครั้ง หน่วยเป็นมิลลิวินาที
- Backoff เป็นปัจจัยในการบอกว่า การทดลองใหม่ในแต่ละครั้งนั้น (ซึ่งจะอยู่ในช่วง n และ n+1) จะให้เพิ่มช่วงหน่วงเวลาเท่าไร คำนวณได้จาก delay time ยกกำลัง n+1 เช่น หากพิจารณาจากรูปที่ 2.6 กำหนดค่า delay time = 1 วินาที และ Backoff =2 จะคำนวณได้ว่า

- รอบที่ 1 = 1000 x 2^0 = 1000 มิลลิวินาที = 1 วินาที
- รอบที่ 2 = 1000 x 2^1 = 2000 มิลลิวินาที = 2 วินาที
- รอบที่ 3 = 1000 x 2^2 = 4000 มิลลิวินาที = 4 วินาที

➢ ภาวะวิกฤต (Critical) เป็นการบอกว่า หากโปรเซสเซอร์ใดที่ถูกกำหนดให้มีภาวะวิกฤต หากโปรเซสเซอร์นั้นทำงานล้มเหลว ก็ให้หยุดการทำงานทั้งหมด แต่หากไม่ได้กำหนดไว้ หากโปรเซสเซอร์นั้นล้มเหลว เวิร์คโฟลว์ทั้งหมดจะยังคงทำงานได้ต่อไป แต่ส่วนที่เป็นดาวน์สตรีม (downstream) จากโปรเซสเซอร์ที่ล้มเหลวนี้จะไม่ถูกเรียกทำงาน



รูปที่ 2.8 กำหนดค่าการคงทนต่อการทำงาน

- **โหนดแสดงการเชื่อมโยงของข้อมูล**

โหนดแสดงการเชื่อมโยงของข้อมูล (Data Link Nodes) ที่ปรากฏใน AME คือ ข้อมูลแสดงการเชื่อมต่อของแต่ละโหนดในเวิร์คโฟลว์ โหนดเหล่านี้จะเกิดโดยอัตโนมัติเมื่อผู้ใช้ทำการต่ออินพุตและเอาท์พุตของโหนดที่เกี่ยวของในเวิร์คโฟลว์นั้นๆ การลบโหนดใดโหนดหนึ่งก็จะหมายถึงการยกเลิกการเชื่อมต่อของข้อมูลที่โหนดนั้นๆ

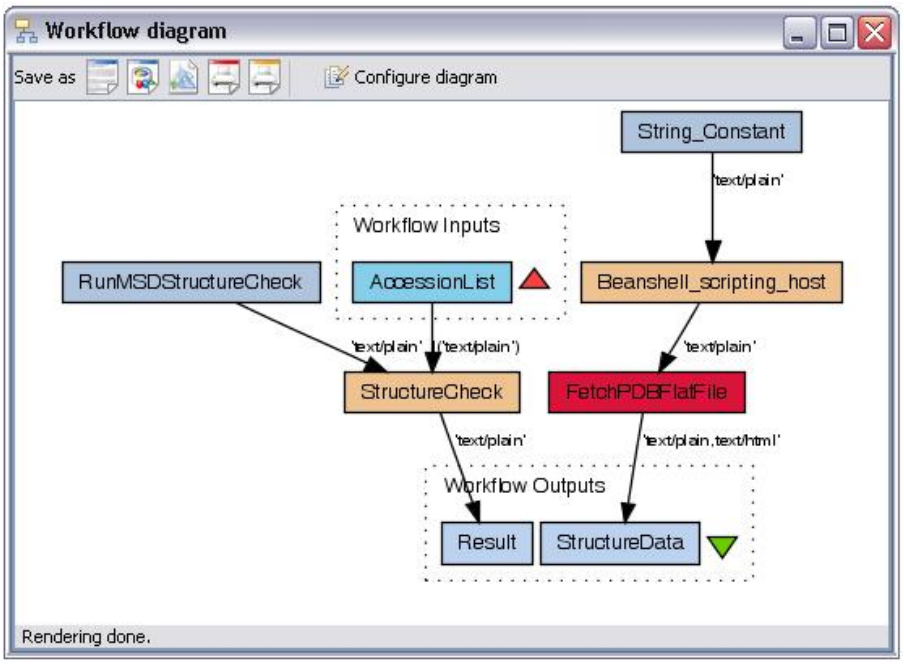
- **โหนดแสดงการควบคุมการเชื่อมต่อ**

โหนดแสดงการควบคุมการเชื่อมต่อ (Control link nodes) แสดงรายละเอียดการทำงานของโปรเซสเซอร์ที่ขึ้นอยู่กับเงื่อนไขนั้นๆ โหนดเหล่านี้จะเกิดโดยอัตโนมัติเมื่อผู้ใช้ทำการ

กำหนดเงื่อนไขต่างๆในเวิร์คโฟลว์ การลบโหนดใดโหนดหนึ่งก็จะหมายถึงการยกเลิกการควบคุมการเชื่อมต่อที่โหนดนั้นๆ

2) ไดอะแกรมของเวิร์คโฟลว์

ไดอะแกรมของเวิร์คโฟลว์ (Workflow diagram) เป็นส่วนประกอบที่ทำหน้าที่แสดงผลเพียงอย่างเดียวของเวิร์คโฟลว์ โดยอินพุต, เอาท์พุต, และโปรเซสเซอร์จะแสดงในรูปของกล่องสีต่างๆ และลูกศรแทนการเชื่อมต่อข้อมูล (Data links) และการควบคุมการเชื่อมต่อ (Control links) ดังรูปที่ 2.9



รูปที่ 2.9 ไดอะแกรมของเวิร์คโฟลว์

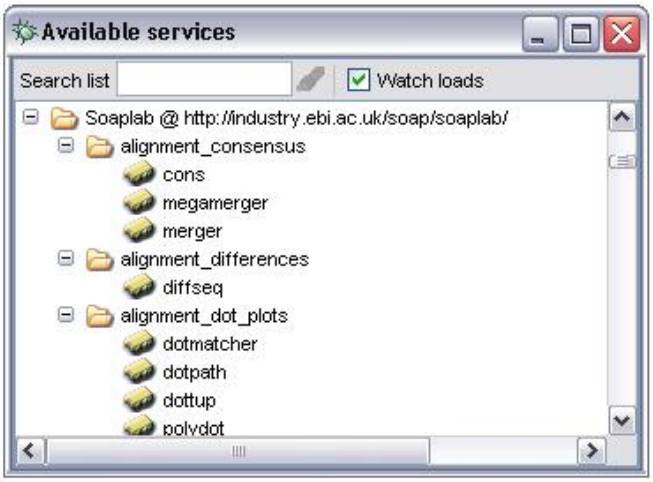
3) บริการที่พร้อมบริการ

เมื่อเริ่มต้นโปรแกรมทาเวอร์นา โปรแกรมจะค้นหาบริการที่สามารถเข้าถึงได้และพร้อมให้บริการ (Available services) ทั้งหมดโดยโปรแกรมจะทำการตรวจสอบการตอบกลับ (Responding) ของเอ็นพอยต์ (Endpoint) ของบริการต่างๆด้วย หากบริการใดที่ไม่มีการตอบกลับก็จะไม่ปรากฏในรายชื่อบริการที่พร้อมให้บริการ ขณะเดียวกันเอ็นพอยต์ของบริการใดที่พร้อม

บริการหรือมีการตอบกลับ บริการเหล่านี้จะเป็นบริการที่ทันสมัยที่สุด ณ เวลาที่ผู้ใช้งานโปรแกรมตอนนั้นๆ ตัวอย่างดังรูปที่ 2.10 แสดงโหนดบริการต่างๆของ Soaplab จาก EBI และค่าเอ็นพอยต์โดยปริยายกำหนดไว้ในไฟล์ *mygird.properties* โดยค่าโดยปริยายจะมีดังนี้

- เว็บเซอร์วิสที่อยู่บนพื้นฐานของมาตรฐาน Web Services Description Language (WSDL)
- Soaplab service คือ เว็บเซอร์วิสที่มีวิธีการการสร้างด้วยโปรแกรม Soaplab Analysis Tool Analysis Tool ซึ่งพัฒนาจาก EBI
- Biomoby คือเว็บเซอร์วิสจาก Biomoby
- BioMart คือเว็บเซอร์วิสจาก BioMart

นอกจากนี้ ผู้ใช้งานสามารถเพิ่มเอ็นพอยต์ที่ต้องการเข้ามาในโปรแกรมทาเวอร์นาได้ด้วยตนเองอีกด้วย เช่นในกรณีที่ผู้ใช้งานสร้างบริการขึ้นมาใช้งานเอง และสร้างระบบเชื่อมประสานให้โปรแกรมทาเวอร์นาเข้าถึงบริการนั่นได้ ซึ่งวิทยานิพนธ์นี้ก็จะนำเสนอวิธีการสร้างและเข้าถึงบริการเหล่านี้เช่นกัน โดยจะกล่าวรายละเอียดต่อไป

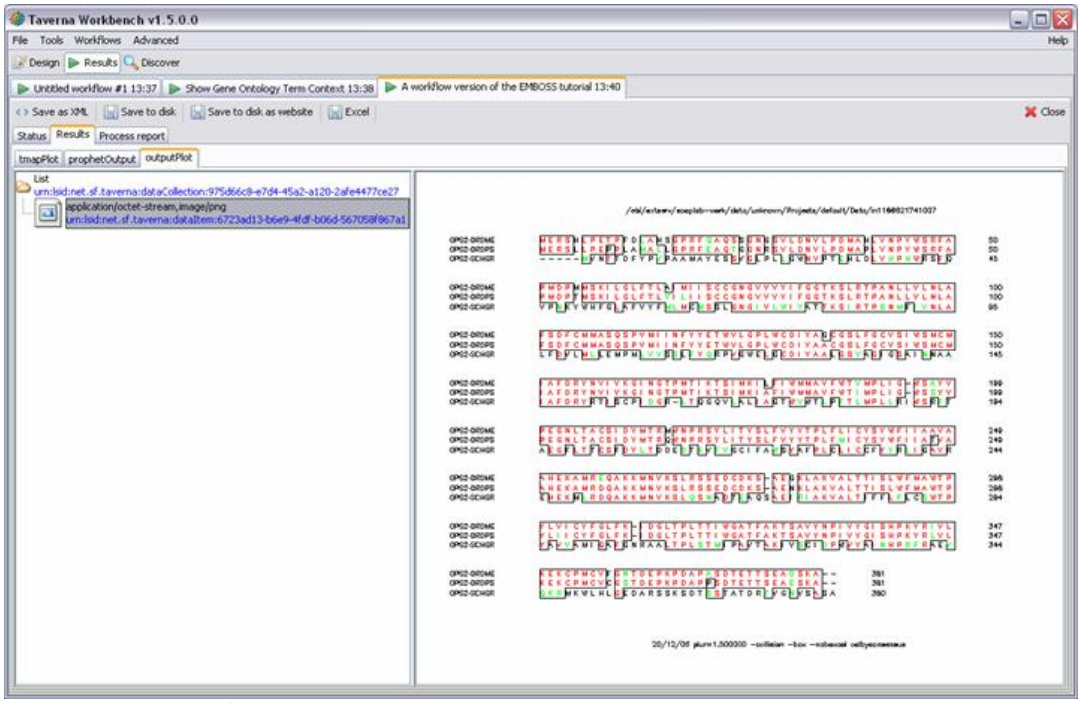


รูปที่ 2.10 *Available Service Panel* แสดงโหนดของบริการ Soaplab จาก EBI

4) หน้าต่างแสดงผลการทำงาน

หน้าต่างแสดงผลการทำงาน (Result browser) ทำหน้าที่แสดงผลการทำงานเมื่อทุกโปรเซสเซอร์ในเวิร์คโฟลว์ทำงานเสร็จแล้ว หรือหากบางโปรเซสเซอร์ในเวิร์คโฟลว์ล้มเหลวใน

ระหว่างการทำงาน แต่ถ้ามีบางโปรเซสเซอร์ที่สามารถทำงานได้สำเร็จก็จะแสดงผลได้เช่นเดียวกัน นอกจากนี้ยังแสดงสถานภาพทำงานและรายงานโปรเซส (Process report) สำหรับดูสถานะและรายงานสรุปการทำงานด้วย ดังรูปที่ 2.11 ผลลัพธ์เป็นกราฟิกชนิดข้อมูล PNG ซึ่งสามารถกำหนดชนิดของเอาท์พุตได้จาก AME ตาม MIME Type ที่โปรแกรมทาเวอร์นาสนับสนุนไว้



รูปที่ 2.11 ผลลัพธ์การทำงานของเวิร์คโฟลว์จาก Result Browser

นอกจากนี้ทาเวอร์นายังมีตัวเรนเดอร์เอาท์พุต (Renderer) อื่นๆ ที่สนับสนุนอาทิ เช่น SeqVISTA [43] ซึ่งสนับสนุน MIME Type ดังนี้ Chemical/seq-aa-genpept, Chemical/seq-na-genbank, Chemical/x-swissprot, Chemical/x-embl-dl-nucleotide และ Chemical/x-ppd

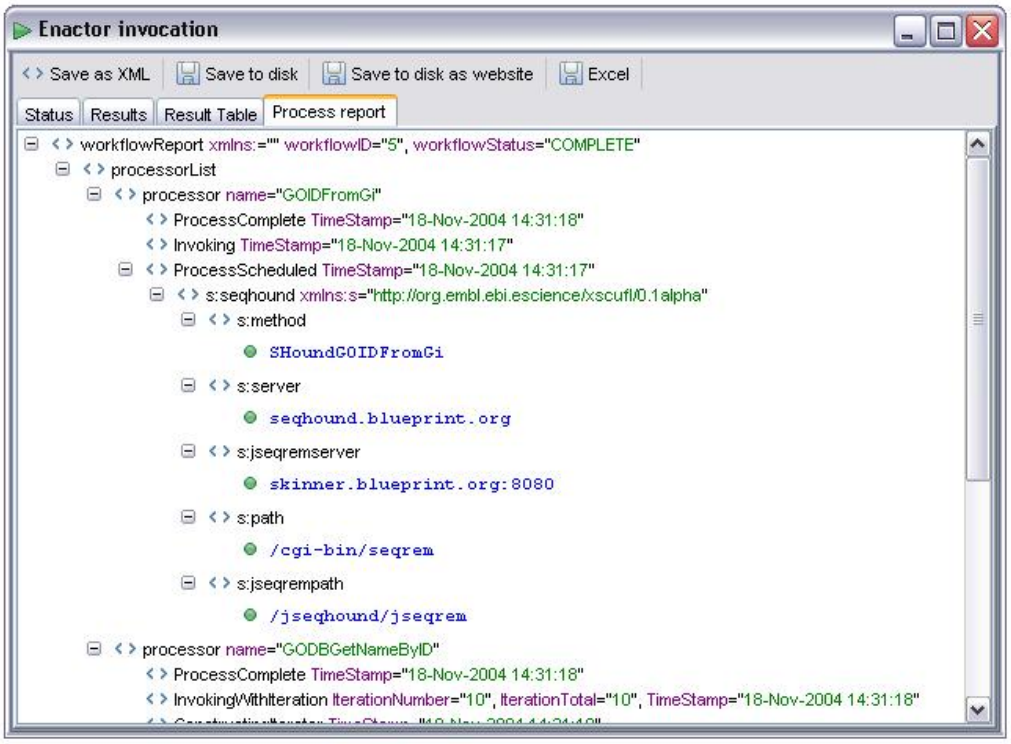
การบันทึกผลลัพธ์การทำงานในโปรแกรมทาเวอร์นามีหลายตัวเลือก ซึ่งแต่ละชนิดบันทึกผลลัพธ์จะแตกต่างกันไปตามลักษณะของรูปแบบที่เลือกดังนี้

- *Saving Entire Result Set as XML*: เป็นการบันทึกผลลัพธ์ในแต่ละรายการให้เป็นภาษา XML
- *Saving Entire Result Set as a Set of Files*: เป็นการบันทึกผลลัพธ์ทั้งหมดของ Workflow ลงดิสก์ ตามค่าชนิดของข้อมูลที่ได้กำหนดใน AME

- *Exporting Results to Excel*: เป็นการบันทึกข้อมูลชนิดข้อความ (Text) เป็นไฟล์ในรูปแบบของ Microsoft Excel ซึ่งจะช่วยในการจัดการข้อมูลต่อไป

- **รายงานโปรเซส**

รายงานโปรเซส (Process report) แสดงบันทึกการทำงานหรือล็อก (Log) การทำงานของเวิร์คโฟลว์เพื่อแสดงว่า เวิร์คโฟลว์ทำสำเร็จหรือไม่ อย่างไร โดยบันทึกรายละเอียดของทุกโปรเซสเซอร์ ซึ่งเป็นการช่วยในการดีบัก (Debug) หรือตรวจสอบได้ทางหนึ่งว่าแต่ละโปรเซสเซอร์ว่าทำงานอย่างไร ใช้เวลาเท่าไร หรือล้มเหลวในช่วงไหนของการทำงาน ตัวอย่างดังรูปที่ 2.12



รูปที่ 2.12 รายละเอียดการทำงานของรายงานโปรเซส

2.3 Scufl และลักษณะเด่นของโปรแกรมทาเวอร์นา

Scufl ย่อมาจาก Simple Conceptual Unified Flow Language [2][4] เป็นภาษาและมาตรฐานของโปรแกรมทาเวอร์นาที่โครงการมายกริดสร้างขึ้นมา คุณสมบัติเด่นๆของภาษานี้มีดังนี้

2.3.1 การทำงานซ้ำๆ โดยปริยาย (Implicit iteration)

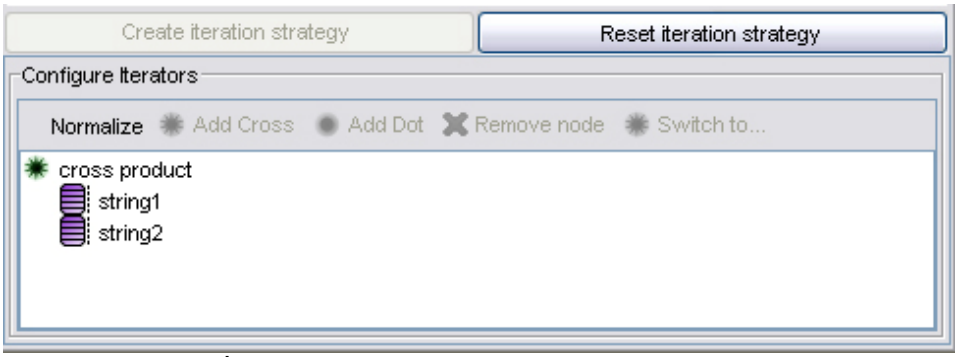
Iteration หรือการทำงานซ้ำ ในบริบทนี้คือการทำงานที่แบบซ้ำๆของกระบวนการใดการหนึ่งกับข้อมูลหลายรายการ ยกตัวอย่างเช่น กระบวนการทำงานการวิเคราะห์ Guanine-cytosine content (GC) จากเซ็ตของลำดับพันธุกรรมหรือลำดับเบส ในภาษาการทำงานแบบดั้งเดิมอย่างเช่นภาษา Perl นั้น สามารถจัดการสิ่งเหล่านี้ได้ด้วยการทำเป็นลูป แล้วระบุเงื่อนไขการทำงานในลูปนั้นตัวอย่างเช่น *'while <condition> do <something>'* โดยการทำงานจะอยู่ภายใต้เงื่อนไขที่ระบุตราบใดที่ยังคงมีรายการข้อมูลให้เหลือไว้ประมวลผล

สิ่งแวดล้อมของโปรแกรมทาเวอร์นา นำเสนอกรอบการทำงานอย่างง่ายในการจัดการเกี่ยวกับลูป ผู้ใช้งานเพียงแต่ต่อเอาท์พุตซึ่งบรรจุเซ็ตของรายการข้อมูลเข้ากับอินพุตซึ่งรับข้อมูลทีละหนึ่งรายการได้เลย จากตัวอย่างการวิเคราะห์ GC ผู้ใช้อาจจะต้องการใช้บริการแค่หนึ่งบริการซึ่งใช้ข้อมูลลำดับเบส (DNA) หนึ่งลำดับ และผลิตเอาท์พุตเป็นเลขทศนิยมบอกถึงระดับความเข้มข้นของ GC ของลำดับนั้นๆ ในโปรแกรมทาเวอร์นา ผู้ใช้สามารถต่อเซ็ตของลำดับต่างๆเข้าไปที่อินพุตเดียวกันได้เลย โดยโปรแกรมจะทำงานครั้งละหนึ่งลำดับจนข้อมูลอินพุตหมดสิ้น จากเอาท์พุตปกติคือเลขทศนิยมค่าเดียวต่อหนึ่งลำดับก็ไปเป็นรายการของเลขทศนิยม โดยรายการแรกของเอาท์พุตก็จะตรงกับรายการแรกของอินพุต เอาท์พุตที่สองก็จะตรงกับอินพุตที่สองแบบนี้ไปเรื่อยๆ

- **การทำงานซ้ำๆกับหลายอินพุต (Implicit iteration over multiple inputs)**

การทำงานโดยปริยายก็คือ ค่าต่างๆของอินพุตจะผสมผสานกันทั้งหมด จากรูปที่ 2.13 แสดงกลไกของการทำงานซ้ำกันของบริการหนึ่งแบบง่ายๆ โดยเป็นการเชื่อมต่อสายอักษร

สองสายเข้าด้วยกัน ในโปรแกรมทาเวอร์นาสามารถเห็นการกำหนดค่าการซ้ำกันของข้อมูลโดยปริยายคือ หลายพอร์ตต่อหลายพอร์ต (All against all)

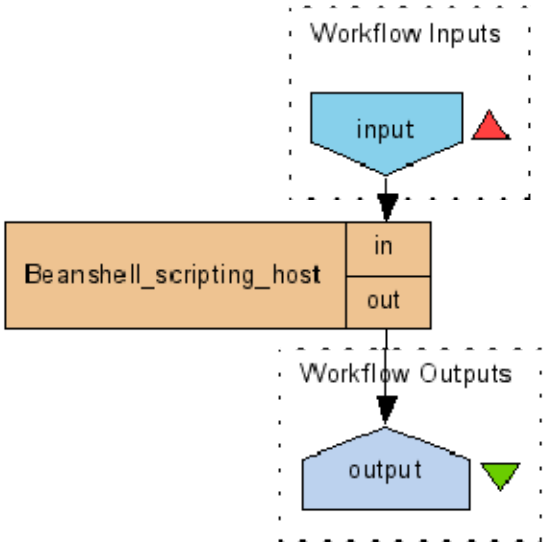


รูปที่ 2.13 Configure iterators ในโปรแกรมทาเวอร์นา

กลไกของการซ้ำกันแสดงแบบโครงสร้างต้นไม้ โหนดใบของต้นไม้ตอบสนองต่ออินพุตนั้นๆให้กับบริการที่เชื่อมต่อเข้าด้วยกัน ในกรณีนี้คือ *String1* และ *String2* และที่ไม่ใช่โหนดข้อมูลคือวิธีการผสมผสานกันของอินพุต แบ่งเป็นสองชนิดคือ Cross product คือการผสมกันของพอร์ตข้อมูลแบบทั้งหมดต่อทั้งหมด (All against all) ซึ่งสามารถเปลี่ยนเป็น ลำดับแรกกับลำดับแรก ลำดับสองกับลำดับที่สอง แบบนี้ไปเรื่อยๆ และอย่างไรก็ตาม กลไกการปรับแต่งการซ้ำของข้อมูลสามารถกำหนด Cross product และ Dot product แบบผสมกันได้ตามลักษณะการทำงานของเวิร์คโฟลว์ ซึ่งในวิทยานิพนธ์จะใช้แบบผสมกัน

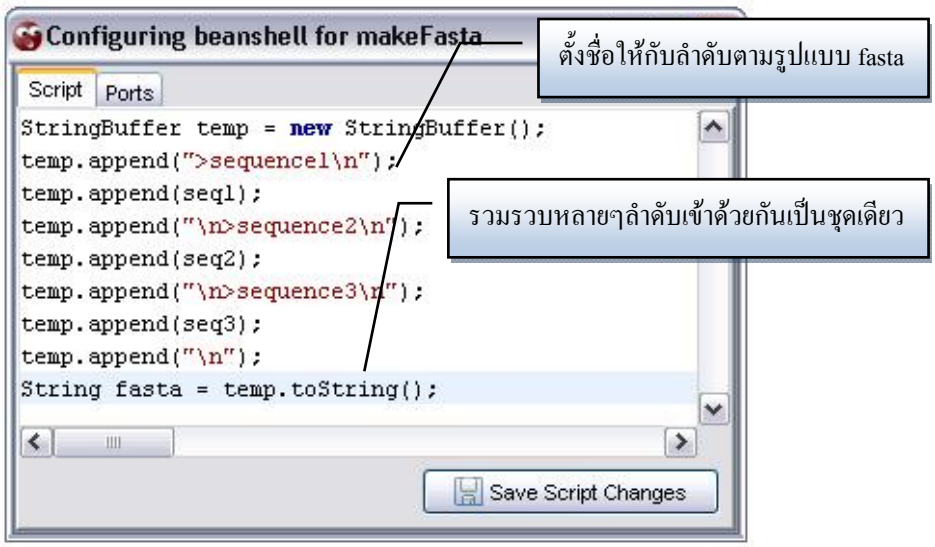
2.3.2 บีนเชลล์สคริปต์ (Beanshell scripting)

โดยปกติแล้วเอาท์พุตของบริการหนึ่งๆ อาจไม่ได้มีรูปแบบ (Format) ที่เหมาะสมและถูกต้องสำหรับบริการถัดไปหรือสำหรับ Nested workflow ได้ จึงต้องมีทางเลือกที่สามารถแก้ปัญหากรณีเหล่านี้ โดยบริการที่เข้ามาทำหน้าที่ดังกล่าวนี้เรียกว่า *Shim Service* [9] หรือเป็นบริการที่ไม่ซับซ้อนมาก ทำหน้าที่เปลี่ยนแปลงโครงสร้างหรือรูปแบบข้อมูลตลอดไปจนถึงการรองรับการเขียนภาษาโปรแกรมเพื่อรองรับอัลกอริทึมจากผู้ใช้งานเอง รูปที่ 2.14 แสดงเวิร์คโฟลว์การใช้งานบีนเชลล์อย่างง่ายโดยหมายถึงผลเอาท์พุตคืออินพุต (Output = Input)



รูปที่ 2.14 เวิร์คโฟลว์การใช้งานบีนเชลล์อย่างง่าย

บีนเชลล์สคริปต์ (Beanshell scripts) จัดเป็นบริการท้องถิ่นในโปรแกรมทาเวอร์นา (Java local service) ที่ใช้ในการแก้ปัญหานี้ โดยผู้ใช้สามารถเขียนภาษาสคริปต์เพื่อเข้าถึงตัวแปล คำสั่งของภาษา Java หรือ Java interpreter โดยไม่ต้องอาศัยความรู้พื้นฐานด้านการโปรแกรมภาษา Java มากนัก จากรูปที่ 2.15 เป็นตัวอย่างของสคริปต์ในการปรับแต่งสายอักษรของลำดับเบสซึ่งเป็นอินพุตให้กับบีนเชลล์ ซึ่งรวบรวมไปเป็นชุดข้อมูลเอาท์พุตแบบ Fasta (Fasta format) [36] โดยรวบรวมหลายๆลำดับเบสเข้าไว้ด้วยกัน ซึ่งในวิทยานิพนธ์นี้ใช้บีนเชลล์ในการสร้างสคริปต์เพื่อตรวจสอบบริการไบโอมาร์ท

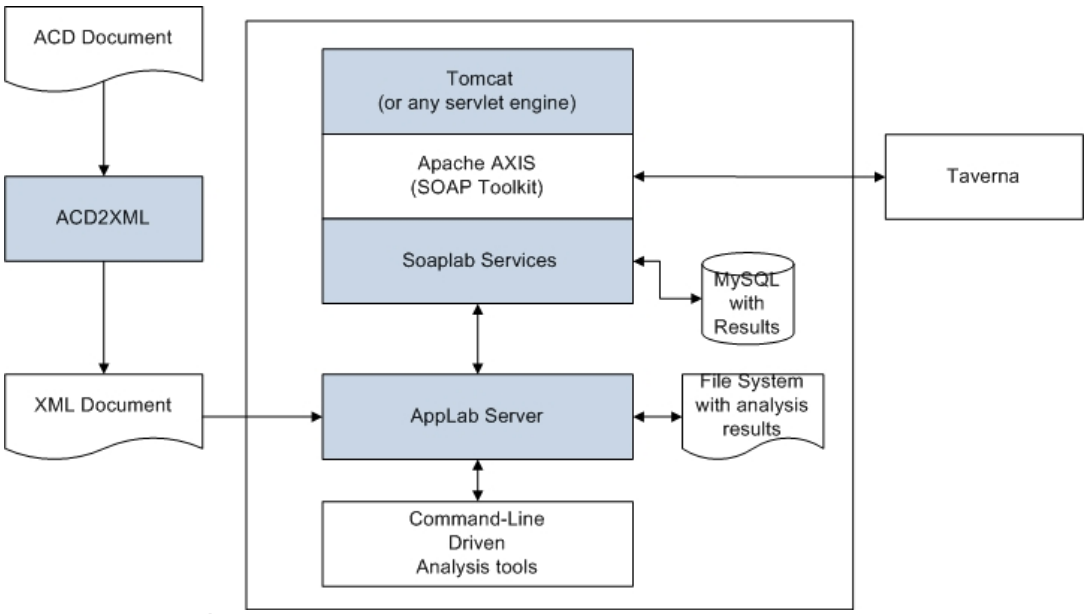


รูปที่ 2.15 ตัวอย่างสคริปต์ของบีนเชลล์

2.3.3 บริการ Soaplab

Soaplab Analysis Tool [37] คือกรอบการทำงานแบบหนึ่งที่สามารถทำให้โปรแกรมประยุกต์ที่อยู่ในรูปแบบของ Command-line สามารถทำงานเป็นเว็บเซอร์วิสได้ ส่วนที่สำคัญก็คือตัวระบบที่เป็นที่อยู่จัดการการทำงานของเว็บเซอร์วิส ในรูปที่ 2.16 คือสถาปัตยกรรมการทำงานของ Soaplab Analysis Tool จากรูป Tomcat คือระบบที่เป็นที่อยู่และตัวจัดการการทำงานของเว็บเซอร์วิส บริการ Soaplab ก็คือรายชื่อของเว็บเซอร์วิสต่างๆ ซึ่งมีหรือพอยน์เตอร์ (pointer) เชื่อมโยงไปยัง AppLab server ซึ่งเป็นระบบที่ครอบหรือห่อโปรแกรมประยุกต์ที่อยู่ในรูปแบบของ Command-line

Soaplab Analysis Tool เป็นคอมโพเนนท์ส่วนหนึ่งของมายกริดในโปรแกรมทาเวอร์นา โดยมี SOAP Toolkit เป็นตัวเชื่อมประสานระหว่างกันซึ่งการติดต่อสื่อสารก็จะเป็นไปตามโปรโตคอล SOAP ในที่นี้คือ Apache AXIS [45] จะทำงานเป็น SOAP Toolkit



รูปที่ 2.16 สถาปัตยกรรมการทำงานของ Soaplab Analysis Tool [37]

2.3.4 การดีบักและชี้นำทิศทางการทำงานของเวิร์คโฟลว์ด้วยเบรคพอยต์

ประโยชน์ของเวิร์คโฟลว์อีกอย่างหนึ่งคือ สามารถควบคุมทิศทางการทำงานได้ โดยในหัวข้อนี้จะอธิบายการชี้นำหรือการควบคุมทิศทางการทำงานของเวิร์คโฟลว์คือ การหยุดการทำงาน และแก้ค่าของตัวแปรต่างๆระหว่างทำงาน ซึ่งมีการติดต่อกับเวิร์คโฟลว์สองแบบคือ การหยุดชั่วคราว (Pausing) และการยกเลิก (Cancelling)

การหยุดการทำงานของเวิร์คโฟลว์สามารถทำได้สองแนวทางคือ โดยการใส่เบรคพอยต์ (Breakpoints) ที่จุดใดจุดหนึ่งของเวิร์คโฟลว์ซึ่งจะหยุดโดยอัตโนมัติ หรือจะหยุดการทำงานของเวิร์คโฟลว์เองทั้งหมดโดยการสั่งเองโดยผู้ใช้งาน (Manual) [2][4]

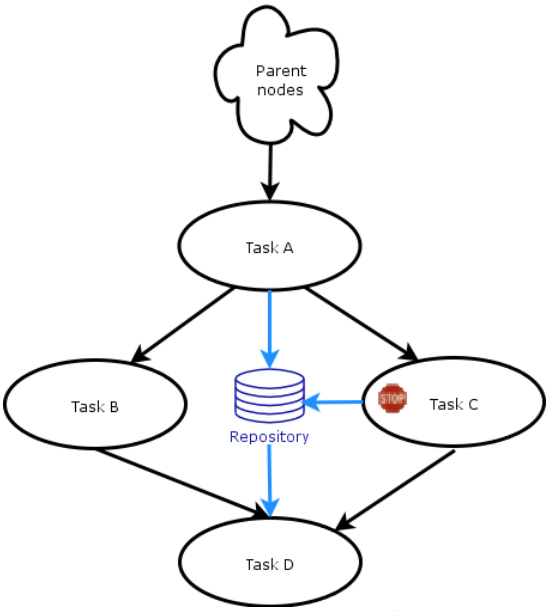
- **เบรกพอยต์ (Breakpoints)**

เบรคพอยต์เป็นศัพท์เทคนิคมาจากเครื่องมือการดีบักซอฟต์แวร์ เครื่องมือเหล่านี้จะทำจุดหรือเครื่องหมาย (Mark) ในรหัสของโปรแกรม เพื่อให้โปรแกรมหยุดการทำงานให้ผู้ใช้งานสามารถเข้ามาแทรกแซงการทำงานของโปรแกรมได้ ในความหมายเดียวกัน เบรกพอยต์ใน

เวิร์คโฟลว์จะทำจุดหรือเครื่องหมายที่บริการหรือโปรเซสเซอร์ เพื่อให้หยุดการทำงาน ข้อมูลอินพุตและเอาท์พุตที่หยุดชั่วคราวที่บริการนั้นอาจจะแก้ไขหรือเปลี่ยนค่าได้

- **การแก้ไขข้อมูลระหว่างบริการหรือโปรเซสเซอร์**

การทำงานของเบรกพอยต์ในโปรแกรมทาเวอร์นา จะตรงข้ามกับการดีบักในวิศวกรรมซอฟต์แวร์ กล่าวคือเบรกพอยต์ในโปรแกรมทาเวอร์นาจะหยุดการทำงานของบริการก็ต่อเมื่อบริการได้ผลิตเอาท์พุตเสร็จเรียบร้อยแล้ว ซึ่งเป็นการหยุดก่อนที่จะเริ่มต้นการทำงานของบริการถัดไป ซึ่งจุดนี้ผู้ใช้สามารถเข้าไปแก้ข้อมูลเหล่านี้ก่อนที่ส่งให้กับบริการถัดไปได้ เหตุผลของกระบวนเหล่านี้คือ ต้องการที่จะรักษาความเป็นหนึ่งเดียวและความต่อเนื่องของข้อมูลในการกระจายและการทำงานแบบขนาน และยังป้องกันกรณีที่ไม่สามารถหยุดการทำงานของเวิร์คโฟลว์ได้ ยกตัวอย่างเช่น จากรูปที่ 2.17 บริการเอมีเอาท์พุตสองเอาท์พุตโดยแต่ละเอาท์พุตส่งต่อไปยังบริการบีและบริการซี ถ้าตัวดีบักแบบทั่วๆไปเพิ่มเบรคพอยต์ที่บริการซีและเปลี่ยนข้อมูลที่บริการเอจะทำให้บริการซีมีข้อมูลใหม่ ในขณะที่บริการบียังมีข้อมูลเดิมอยู่ สถานการณ์แบบนี้เป็นการเพิ่มเงื่อนไข หรือมีแนวโน้มที่จะทำให้เวิร์คโฟลว์ผลิตผลลัพธ์ที่ไม่ถูกต้องหรือตรงตามความต้องการ นอกจากนี้ในขณะที่เวิร์คโฟลว์กำลังทำงาน อินพุตหรือเอาท์พุตจะถูกบันทึกไว้ใน Repository ซึ่งข้อมูลเดียวกันก็จะเก็บทั้งข้อมูลรุ่นเก่าและรุ่นใหม่ ดังนั้นในการแก้ไขหรือตรวจสอบข้อมูลเหล่านี้ที่จุดใดจุดหนึ่งของเวิร์คโฟลว์ ควรจะกำหนดเบรกพอยต์ที่บริการที่กำลังจะผลิตข้อมูล (หรือกำลังจะประมวลผลต่อ) และไม่ใช่บริการที่จะรับข้อมูลนั้นมาเป็นอินพุต



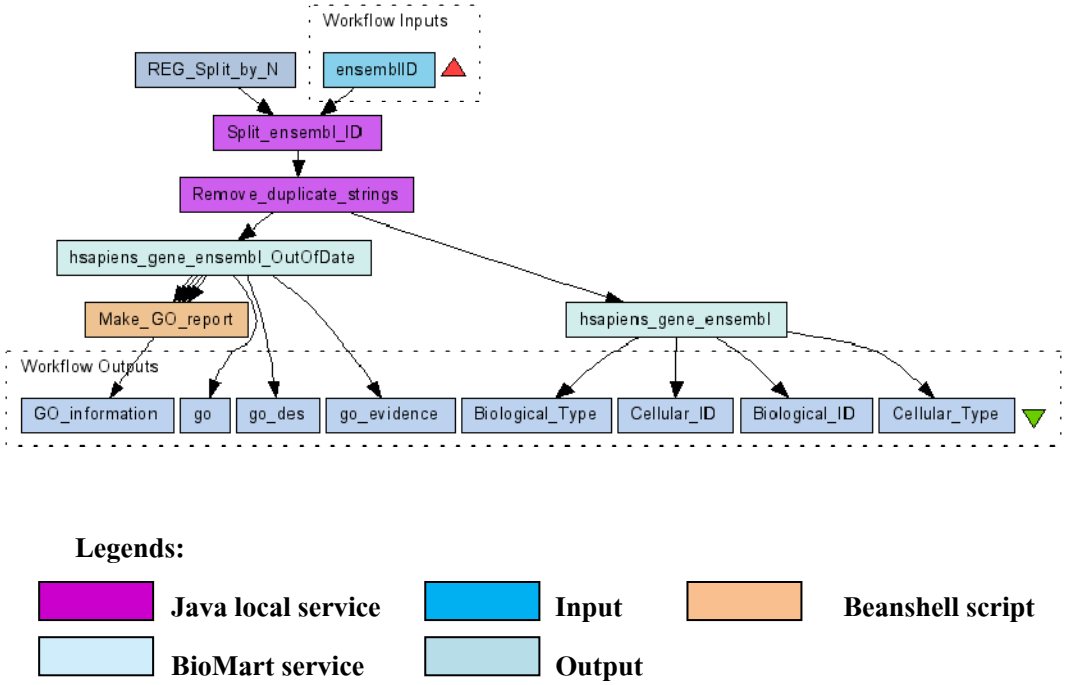
รูปที่ 2.17 ตัวอย่างกลไกการทำงานของเวิร์คโฟลว์เมื่อมีการกำหนดเอ็นพอยต์ [9]

2.4 ทาเวอร์นาสอง รุ่นทดลองใช้งาน (Taverna 2 Preview)

โปรแกรมทาเวอร์นาตั้งแต่รุ่น 1.7.0 ก็มีปลักอินชื่อว่า *Taverna 2 preview* ซึ่งเป็นการทดลองโปรแกรมทาเวอร์นารุ่น 2 เพื่อการทำงานที่เรียกว่า *Workflow health check* หรือการตรวจสอบเสถียรภาพของเวิร์คโฟลว์ [8][9] โดยจะแปลงตัวแบบเอกสาร Scufl แบบเก่าให้เป็นแบบ T2 style แล้วตรวจสอบว่า เอ็นพอยต์ของเว็บเซอร์วิสต่างๆ ในเวิร์คโฟลว์ตอบกลับผ่าน HTTP (Responding) หรือไม่ หากมีการตอบกลับก็คือสามารถเข้าถึงได้ หากไม่มีการตอบกลับ ปัญหาก็อาจจะมาจากเครือข่ายของผู้ใช้งานเองไม่สามารถติดต่อภายนอกได้ หรือไม่ก็มีปัญหาที่ฝั่งของผู้ให้บริการเองอันเนื่องมาจากหลายๆสาเหตุ

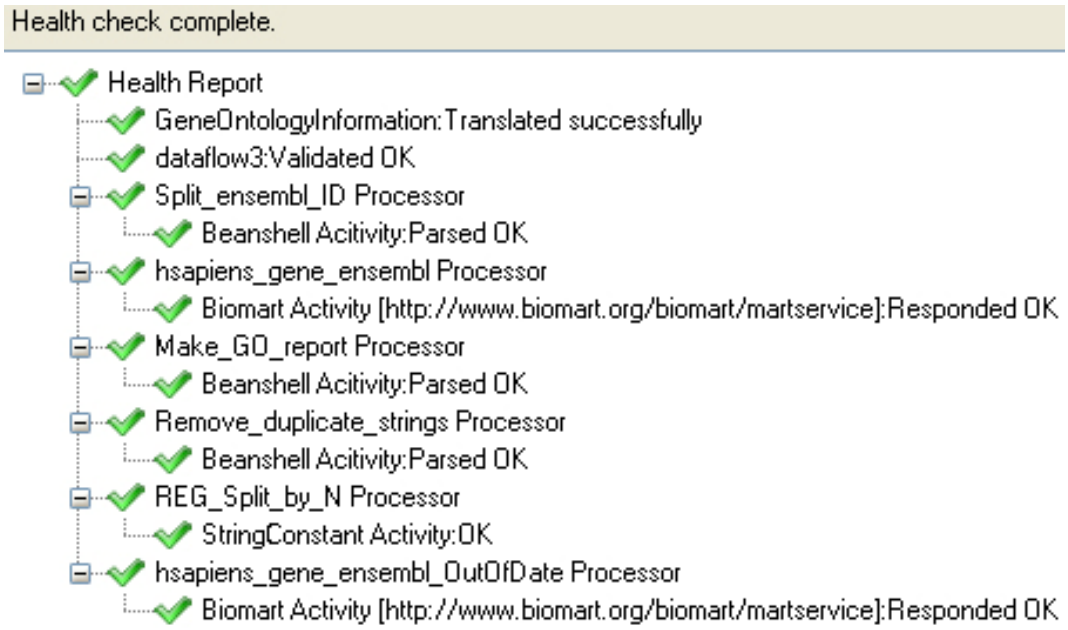
ปลักอิน *Taverna 2 preview* ตรวจสอบเฉพาะการตอบกลับ (Responding) ของเอนพอยต์ของบริการเท่านั้น ไม่ได้ตรวจสอบถึงการเปลี่ยนแปลงอันเนื่องมาจากการปรับปรุง (upgrade) ของบริการนั้นๆด้วย ซึ่งในงานวิจัยนี้เน้นความสนใจไปที่บริการไบโอมาร์ท ซึ่งมีการใช้งานทั้งฟิลเตอร์และแอ็ตตริบิวต์ จากรูปที่ 2.18 แสดงเวิร์คโฟลว์ในการค้นหาข้อมูล Gene Ontology หรือโครงสร้างลำดับการให้คำอธิบายกระบวนการทางชีววิทยา [46] การระบุรายละเอียดในระดับเซล (Cellular localization) และรวมถึงการทำงานในระดับโมเลกุล (Molecular funtion) โดยมีอินพุตคือ Ensembl IDs เวิร์คโฟลว์เรียกใช้บริการไบโอมาร์ทชื่อว่า *hsapiens_gene_ensembl* จำนวน

2 บริการ โดยบริการที่ได้สร้างก่อนหน้านี้และล้าสมัยไปแล้ว และได้ตั้งชื่อบริการที่ล้าสมัยว่า *hsapiens_gene_ensembl_OutOfDate* ซึ่งจะผลิตผลลัพธ์คือ Gene ontology, Gene ontology description และ Gene ontology evidence จากนั้นใช้สคริปต์ของบีนเชลล์ในการเก็บผลลัพธ์เหล่านั้น มารวบรวมเข้าไว้ด้วยกันเพื่อทำเป็นรายงาน

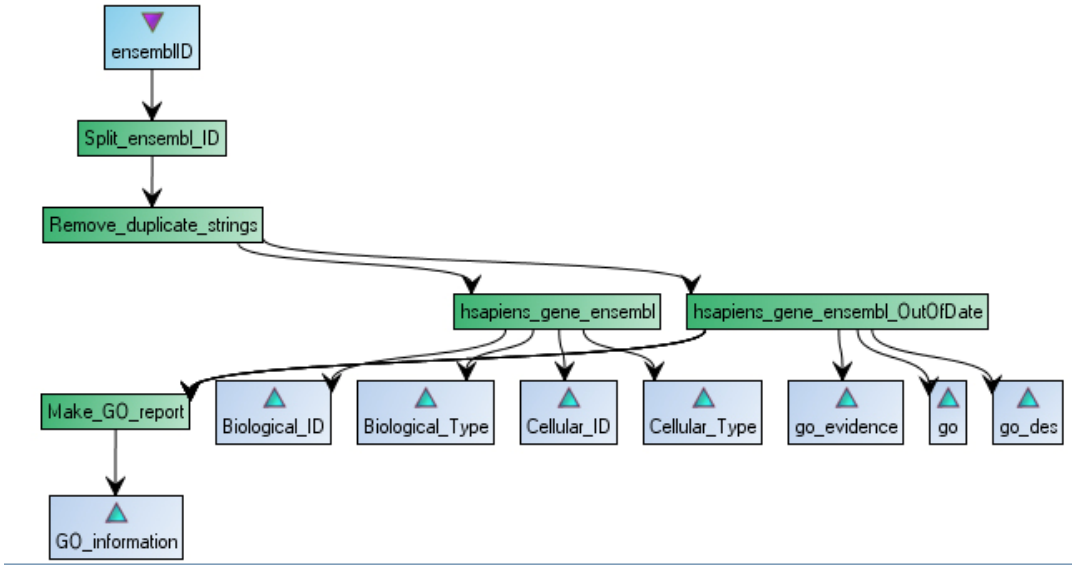


รูปที่ 2.18 เวิร์คโฟลว์ในการค้นหา Gene Ontology จาก Ensembl IDs

เวิร์คโฟลว์ในรูปที่ 2.18 ตรวจสอบด้วย *Taverna 2 preview* และได้ *Health check report* ดังรูปที่ 2.19 ซึ่งแสดงให้เห็นว่า ทุกบริการมีการตอบกลับทั้งสิ้น และในขั้นตอนต่อไปคือเรียกการทำงานของเวิร์คโฟลว์ และได้สถานะการทำงานของเวิร์คโฟลว์ดังรูปที่ 2.20



รูปที่ 2.19 *Health check report* ของเวิร์คโฟลว์



รูปที่ 2.20 สถานะการทำงานของเวิร์คโฟลว์

จากรูปที่ 2.20 บริการสามารถทำงานได้สำเร็จโดยไม่มีข้อผิดพลาดใดๆ แต่จะพบว่า *hsapiens_gene_ensembl_OutOfDate* ไม่ได้ผลผลิตผลลัพธ์ใดๆเลย (empty result list) เนื่องจากแอ็ตตริบิวต์ที่ต้องการคือ Gene ontology, Gene ontology description และ Gene ontology

evidence ได้ล้าสมัยเสียแล้ว อย่างไรก็ตามบีนเชลล์ก็ยังคงทำงานได้สำเร็จ เนื่องจากไม่ได้สนใจความมีอยู่ของข้อมูล (Data availability) เพียงแต่ตรวจสอบเฉพาะความผิดพลาดของไวยากรณ์ขณะที่ทำงานเท่านั้น ดังนั้นในวิทยานิพนธ์นี้จึงได้นำเสนอแนวทางแก้ไขปัญหาดังกล่าวโดยการพัฒนาเวิร์คโฟลว์ที่สามารถตรวจสอบฟิลเตอร์และแอ็ตตริบิวต์ที่ล้าสมัยได้

2.5 บริการไบโอมาร์ท (BioMart)

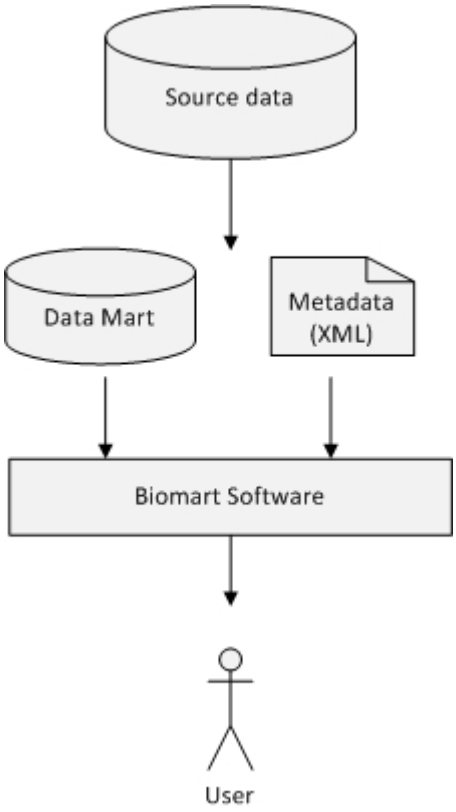
ต่อไปจะกล่าวแนะนำบริการไบโอมาร์ท คำอธิบายบริการ สถาปัตยกรรมการทำงานในภาพรวมของระบบบริการ และการเข้าถึงข้อมูลของบริการไบโอมาร์ทที่นำมาประยุกต์ใช้สำหรับงานวิทยานิพนธ์ มีดังนี้

2.5.1 แนะนำบริการไบโอมาร์ท

เว็บไซต์ของโครงการไบโอมาร์ทคือ http://www.biomart.org ซึ่งสามารถดาวน์โหลดโปรแกรมไบโอมาร์ทรุ่นล่าสุดและเอกสารคู่มือที่เกี่ยวข้องต่างๆได้ โครงการไบโอมาร์ทก่อตั้งร่วมกันโดยสถาบันชีวสารสนเทศแห่งยุโรป (EBI) และศูนย์วิจัยมะเร็ง Cold Spring Harbor [12]

2.5.2 คำอธิบาย

ไบโอมาร์ท คือระบบจัดการข้อมูลแบบ Open source โดยมีระบบเชื่อมประสานให้ผู้ใช้งานสามารถคิวรีข้อมูลชีวสารสนเทศได้หลากหลายและรวดเร็ว [11][12] จุดประสงค์ของไบโอมาร์ทคือ ให้บริการการแปลงข้อมูลใดๆไม่ว่าจะเป็นเอกสารข้อมูลต่างๆหรือฐานข้อมูลไปเป็นข้อมูลที่เรียกว่า ดาตามาร์ท (Data marts) ซึ่งสามารถเข้าถึงด้วยเว็บเบราเซอร์มาตรฐานและยังสามารถเชื่อมต่อกับภาษาโปรแกรม Perl, Java และ API ของเว็บเซอร์วิสอีกด้วย ดังรูปที่ 2.21



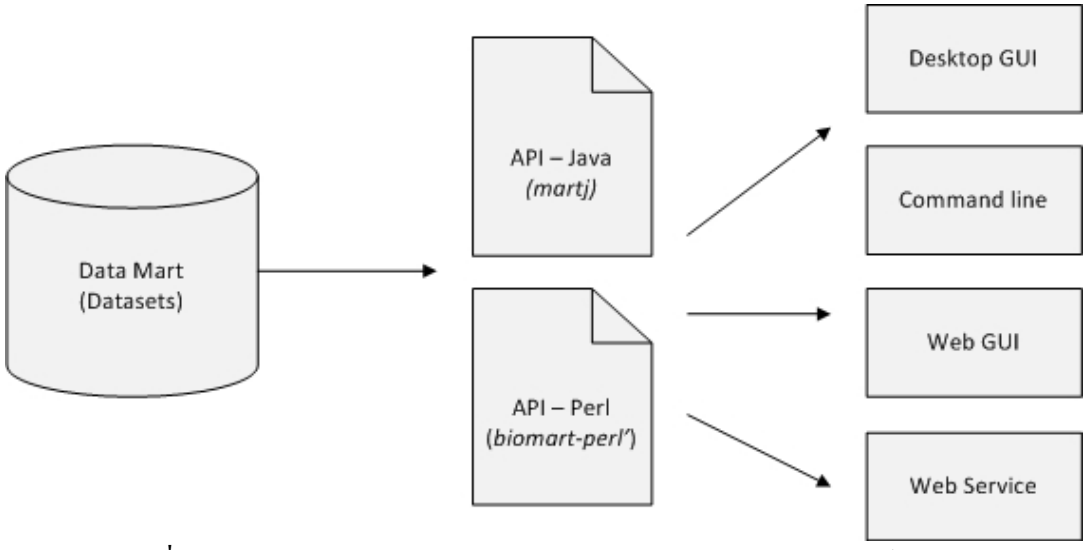
รูปที่ 2.21 ภาพรวมของไบโอมาร์ท [12]

ไบโอมาร์ทสร้างขึ้นเพื่อสนับสนุนการคิวรีข้อมูลได้อย่างดี ผู้ใช้งานสามารถเรียกใช้งานได้อย่างรวดเร็ว, ง่าย, และมีความสามารถในการคิวรีข้อมูลได้อย่างมีประสิทธิภาพ โดยการเลือกเงื่อนไขได้จากเว็บ, กราฟิก, และข้อความ การคิวรีข้อมูลทำได้โดยผ่าน API ของเว็บเซอร์วิสหรือการติดต่อกับไลบรารีโดยการใช้ภาษาโปรแกรม Perl และ Java และชุดโปรแกรมทั้งหมดของไบโอมาร์ทสามารถติดตั้งในเครื่องคอมพิวเตอร์ผู้งานเองได้ด้วย ไบโอมาร์ทเป็นระบบซอฟต์แวร์แบบเปิดภายใต้ลิขสิทธิ์ของ GNU Lesser General Public License (LGPL) [47] ซึ่งทุกๆคนสามารถเข้าถึงและใช้งานโดยไม่มีข้อผูกมัดหรือเงื่อนไขใดๆ

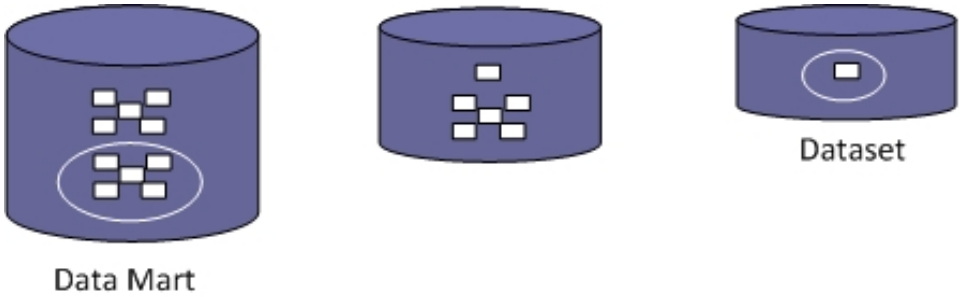
2.5.3 สถาปัตยกรรมของไบโอมาร์ท

ไบโอมาร์ทออกแบบโดยใช้สถาปัตยกรรมแบบ Three Tier Architecture ดังรูปที่ 2.22 โดยส่วนแรกประกอบด้วยฐานข้อมูลเชิงความสัมพันธ์หนึ่งฐานหรือมากกว่า ฐานข้อมูลแต่ละ

ฐานข้อมูลสามารถรองรับได้มากกว่าหนึ่งดาตามาร์ท และในแต่ละดาตามาร์ทจะมีดาตาเซ็ต (Dataset) ได้หลายๆดาตาเซ็ตดังรูปที่ 2.23 การกำหนดการปรับแต่งของดาตาเซ็ตเหล่านี้ จะเก็บอยู่ตารางในแต่ละดาตามาร์ทนั้นๆ ซึ่งสามารถสร้างได้โดยใช้เครื่องมือที่เรียกว่า MartEditor



รูปที่ 2.22 สถาปัตยกรรมแบบ Three Tier Architecture ของไบโอมาร์ท [12]



รูปที่ 2.23 ความสัมพันธ์ระหว่างดาตามาร์ทและดาตาเซ็ต [12]

ส่วนที่สองประกอบด้วย API จำนวนสองชนิดคือ ชนิดที่เขียนขึ้นด้วยภาษา Perl โดยกระจายอยู่ในไลบรารีชื่อ *biomart-perl* และอีกชนิดคือเขียนด้วยภาษา Java โดยกระจายอยู่ในไลบรารีชื่อ *martj*

ส่วนที่สามประกอบด้วยตัวเชื่อมประสานสำหรับงานคิวรีดังนี้

- MartView เป็นตัวเชื่อมประสานที่ทำงานกับเว็บเบราเซอร์ ทำงานบนพื้นฐานของ Perl

- MartService เป็นตัวเชื่อมประสานแบบเว็บเซอร์วิส ทำงานบนพื้นฐานของ Perl
- MartURLAccess เป็นตัวเชื่อมประสานของกลไกการเข้าถึง MartView ผ่านทาง URL ทำงานบนพื้นฐานของ Perl
- MartExplorer เป็นตัวเชื่อมประสานกับผู้ใช้แบบกราฟิก ทำงานบนพื้นฐานของ Java
- MartShell เป็นตัวเชื่อมประสานแบบเชลล์รับคำสั่ง ทำงานบนพื้นฐานของ Java

ข้อกำหนดการปรับแต่งของดาตาเซ็ตจะเก็บในรูปแบบของ XML ในตารางพิเศษของฐานข้อมูลที่ดาตามาร์ทนั้นๆ ผู้ใช้จะจัดการเอกสารรีจีสทรี XML บนไคลเอ็นท์หนึ่งๆโดยจะระบุลงไปว่าดาตาเซ็ตอะไร ในดาตามาร์ทไหนและอยู่ในฐานข้อมูลอะไรที่พร้อมให้บริการสำหรับคิวรีข้อมูล ไบโอมาร์ทรุ่น 0.7 สนับสนุนระบบฐานข้อมูลเชิงสัมพันธ์อย่างเช่น MySQL, Oracle และ Postgres

2.5.4 การเข้าถึงข้อมูลของบริการไบโอมาร์ท

ในการพัฒนาเวิร์คโฟลว์สำหรับตรวจสอบบริการไบโอมาร์ทนี้ ได้ศึกษาถึงกลไกการเข้าถึงข้อมูลที่เกี่ยวข้องของบริการไบโอมาร์ทดังนี้

มาร์ทวิว (MartView) คือตัวเชื่อมประสานทำงานบนเว็บเบราเซอร์และดาตาเซ็ตของไบโอมาร์ทเพื่อให้ผู้ใช้งานสามารถติดต่อสื่อสารกันได้ โดยมาร์ทวิวถูกออกแบบให้ง่ายต่อการใช้งานที่สุดเท่าที่เป็นไปได้ มาร์ทวิวแบ่งตามการใช้งานได้ 2 ประเภทต่อไปนี้

1) การติดต่อด้วยเว็บเบราเซอร์ (Web browser interface)

มาร์ทวิวสามารถเข้าถึงได้ด้วยเว็บเบราเซอร์โดยมี URL ที่ขึ้นอยู่การปรับแต่งของผู้ดูแลระบบที่นำชุดโปรแกรมไบโอมาร์ทมาใช้งานในองค์กรนั้นๆโดยมี URL ศูนย์กลางของมาร์ทวิวที่ใช้ฐานข้อมูลกลางคือ *http://www.biomart.org/biomart/martview*

การใช้บริการไบโอมาร์ทด้วยเว็บเบราเซอร์ จะมีประโยชน์ในการทดสอบการทำงานเบื้องต้นได้ทันที และสามารถตรวจสอบได้ว่าให้ผลลัพธ์ที่ต้องการหรือไม่ นอกจากนี้ ยังใช้วิธีการนี้ในการดีบักหรือตรวจสอบความถูกต้องของบริการอีกด้วย เพราะบริการไบโอมาร์ทที่เข้าถึงผ่านเว็บเบราเซอร์ในขณะนั้นๆย่อมเป็นบริการที่ทันสมัยที่สุด

2) การติดต่อด้วยเว็บเซอร์วิส (Web Services API)

เอพีไอเว็บเซอร์วิสของไบโอมาร์ทมีชื่อว่า มาร์ทเซอร์วิส (MartService) โดยให้บริการเป็นส่วนหนึ่งมาร์ทวิว (MartView) สามารถเข้าถึงได้โดยการร้องขอผ่าน HTTP ซึ่งสามารถใช้เว็บเบราเซอร์มาตรฐานเข้าถึงได้ การเข้าถึงมาร์ทวิวโดยใช้ URL นี้

http://www.mycompany.com/scripts/biomart/martview

เข้าถึงเว็บเซอร์วิสได้ใช้ URL นี้

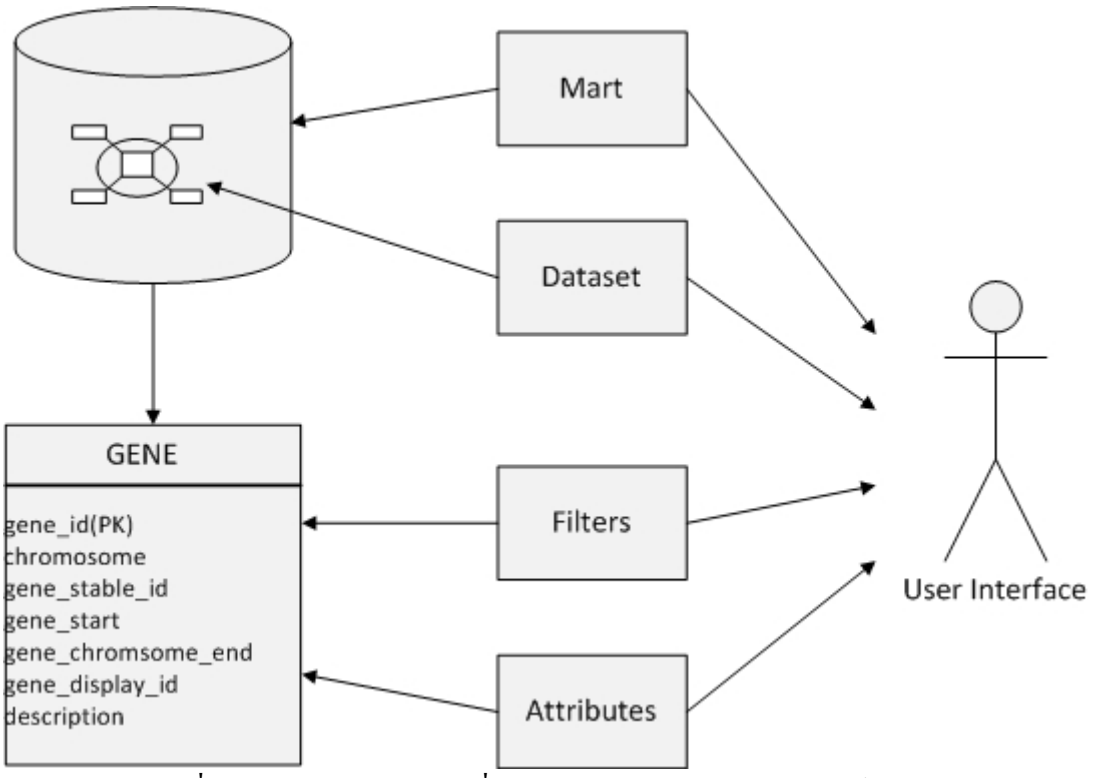
http://www.mycompany.com/scripts/biomart/martservice

และศูนย์กลางเว็บเซอร์วิสไบโอมาร์ทคือ

http://www.biomart.org/biomart/martservice

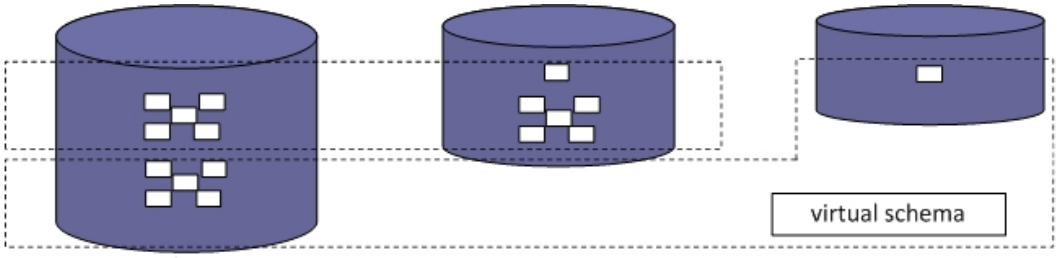
2.1) เมตาดาตา (Metadata)

เมตาดาตา (Metadata) หรือคำอธิบายข้อมูลของเว็บเซอร์วิสต่างๆ สามารถเข้าถึงได้ด้วยการร้องขอ *'GET'* ในขณะที่คำสั่งคิวรีข้อมูลสามารถส่งได้ด้วยการร้องขอ *'POST'* ของ HTTP โดยในดาตาเซ็ตประกอบด้วยฟิลเตอร์หรือฟิลด์ข้อมูลที่ใช้เป็นเงื่อนไขการคิวรีและแอ็ตตริบิวต์หรือฟิลด์ข้อมูลที่เป็นเอาท์พุตจากคิวรีนั้นๆ ตัวอย่างเช่นรูปที่ 2.24 ดาตาเซ็ตของยีนซึ่งประกอบด้วยฟิลเตอร์และแอ็ตตริบิวต์จำนวนมาก ยกตัวอย่างการคิวรีข้อมูลต่อไปนี้คือ หมายเลขยีน (Gene _stable_id), ตำแหน่งเริ่มต้น (Gene_start), ตำแหน่งจบ (Gene_chromosome_end), ชื่อของยีน (Gene_display_id) และคำอธิบาย (Description) โดยมีจะฟิลเตอร์เกี่ยวข้องคือ ระบุการค้นหาจากหมายเลขยีนที่ผู้ใช้เป็นผู้กำหนดเอง และระบุว่าจะค้นที่โครโมโซมไหน เช่น โครโมโซม 1 เป็นต้น



รูปที่ 2.24 ตัวอย่างเมตาดาตาที่ใช้ในการคิวรีข้อมูลจากไบโอมาร์ท [12]

ชุดเมตาดาตาของดาตาเซ็ทที่ให้บริการ สามารถเรียกใช้ได้จากการเติมพารามิเตอร์ต่อท้าย URL พารามิเตอร์แรกก็คือเครื่องหมาย '?' จากนั้นพารามิเตอร์อื่นๆคือเครื่องหมาย '&' ตัวอย่างเช่น .../martservice?type=attributes&dataset=hsapiens_gene_ensembl และต้องระบุ *Virtual schema* ในฐานข้อมูลนั้นๆด้วย (รูปที่ 2.25 ดาตาเซ็ตที่อยู่ภายในเส้นประคือ Virtual schema) การปรับแต่งเมตาดาตาของบริการไอโบมาร์ทดังตารางที่ 2.2 และความหมายของพารามิเตอร์ใช้ในการปรับแต่งดังตารางที่ 2.3



รูปที่ 2.25 ตัวอย่างขอบเขตของ Virtual schema ใดๆของดาตามาร์ท [12]

ตารางที่ 2.2 เมตาดาตาและรายละเอียดการปรับแต่งของบริการไบโอมาร์ท

Metadata	**Type**	**Description**	**virtaulschema**	**mart**	**interface**	**dataset**
Registry file	เอกสารรีริสทรี	เอกสารรีจิสทรีที่ใช้ปรับแต่งการติดตั้งมาร์ทเซอร์วิส	-	-	-	-
Datasets	ดาตาเซ็ต	รายชื่อดาตาเซ็ตที่พร้อมให้บริการ	✓	✓	-	-
Dataset Configuration	การปรับแต่ง	รายละเอียดการปรับแต่งของดาตาเซ็ต	✓	-	✓	-
Attributes	แอ็ตตริบิวต์	ข้อมูลรายละเอียดของแอ็ตตริบิวต์ที่พร้อมบริการของดาตาเซ็ตนั้นๆ	✓	-	✓	✓
Filters	ฟิลเตอร์	ฟิลเตอร์หรือเงื่อนไขการคิวรี	✓	-	-	✓

ตารางที่ 2.3 พารามิเตอร์ที่ใช้ในการปรับแต่งเมตาดาตา

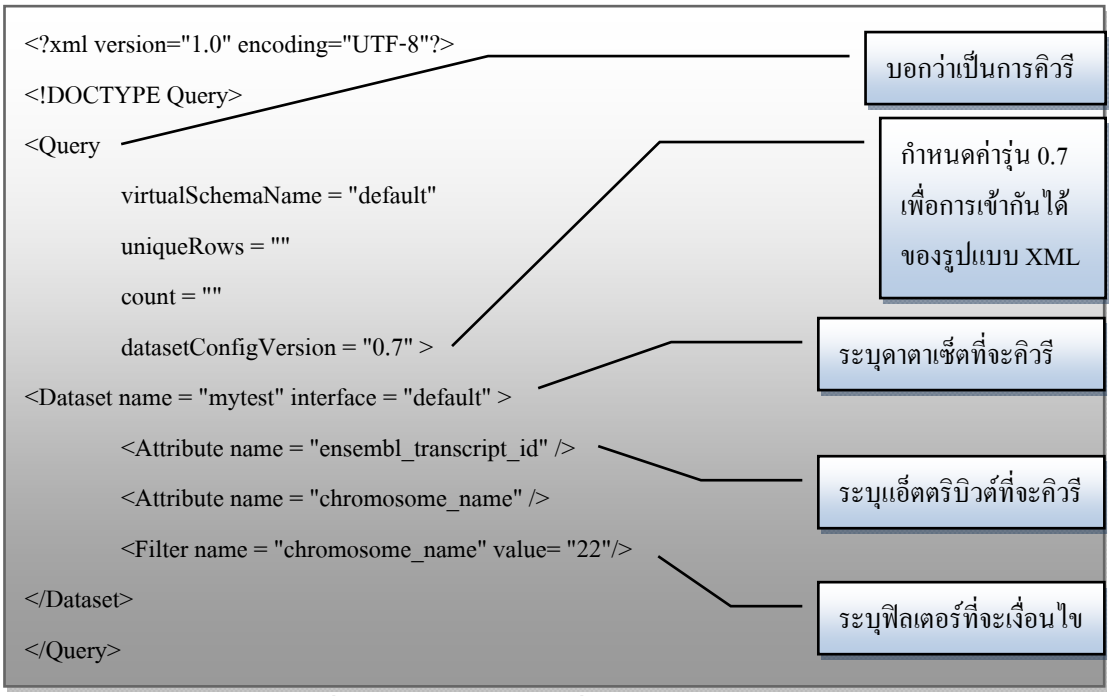
พารามิเตอร์	**ความหมาย**
virtualschema	ชื่อของโครงสร้างข้อมูลที่มีดาตาเซ็ตนั้น (ค่าปริยายคือ *'default'*)
mart	ชื่อของดาตาเซ็ตนั้นๆ
	ชื่อของผู้ใช้งานของดาตาเซ็ต (ค่าปริยายคือ *'default'*)
interface	ชื่อของตัวเชื่อมประสานของดาตาเซ็ตที่ใช้งาน (ค่าปริยายคือ *'default'*)
	ชื่อของผู้ใช้งานของดาตาเซ็ต (ค่าปริยายคือ *'default'*)
dataset	ชื่อของผู้ใช้งานของดาตาเซ็ต (ค่าปริยายคือ *'default'*)

2.2) การคิวรีข้อมูล (Queries)

คำสั่งคิวรีจะถูกส่งให้ระบบด้วยการร้องขอ *'POST'* โดยภายในโครงสร้างคำสั่งคิวรีจะมีพารามิเตอร์ที่ชื่อว่า *query* และค่าพารามิเตอร์ต่างๆ เป็นเอกสาร XML และผลลัพธ์ที่จะได้ก็คือที่ระบุไปเอกสาร XML นี้

- **Query XML syntax**

โครงสร้างเอกสาร XML ดังรูปที่ 2.26 คือตัวอย่างของคำสั่งการคิวรีซึ่งระบุต้องการ 1 ฟิลเตอร์ และ 2 แอ็ตตริบิวต์ ในการคิวรี 1 ดาตาเซ็ต



รูปที่ 2.26 ตัวอย่างของคำสั่งการคิวรีในดาตาเซ็ต

แท็ก *Query* ประกาศตัวเองว่าเป็นคิวรีข้อมูลแท็ก *datasetConfigVerion* กำหนดค่ารุ่น 0.7 เพื่อการเข้ากันได้ของรูปแบบ XML ที่กำหนดค่าการปรับแต่งต่างๆในดาตาเซ็ต แท็ก *Attribute* และ *Filter* สามารถกำหนดได้หลายค่าตามความต้องการของผู้ใช้ โดยแท็ก *Filter* สามารถกำหนดค่าได้ด้วย

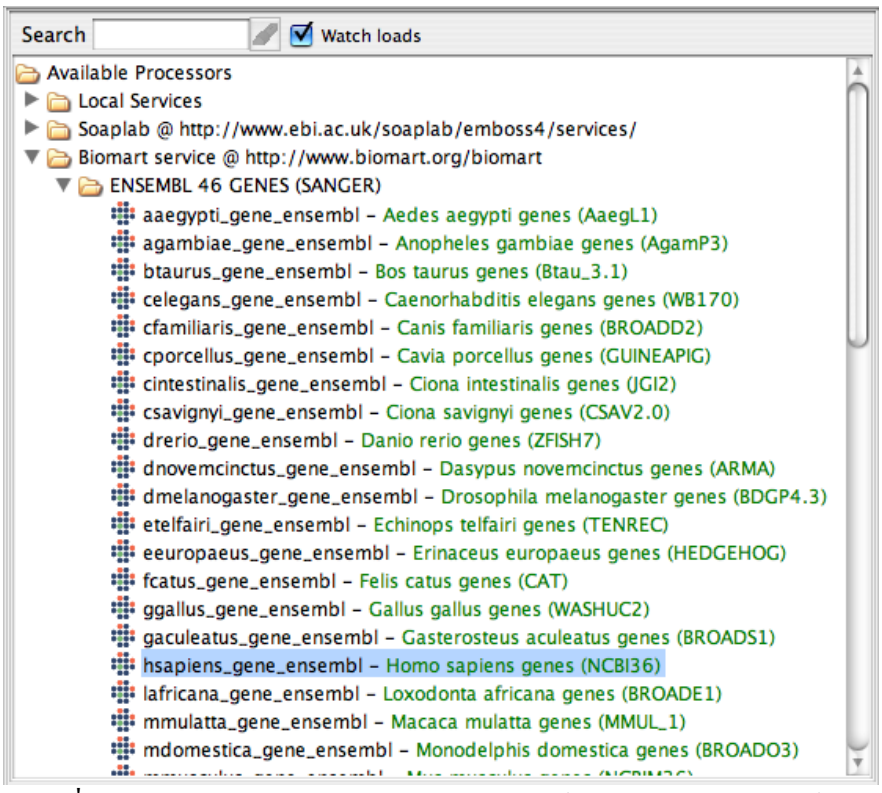
แอ็ตตริบิวต์ที่เป็นผลลัพธ์จะถูกแสดงตามลำดับโครงสร้างข้างต้น โดยมีแท็บเป็นตัวแยกโดยค่าปริยาย ถ้าแท็ก *uniqueRows* ถูกกำหนดค่าให้มีค่าเป็น 1 มาร์ทวิวก็จะกำจัดข้อมูลที่ซ้ำซ้อนกันให้ด้วย ขณะเดียวหากกำหนดค่าแท็ก *count* ให้มีเป็น 1 มาร์ทก็จะทำงานเพียงแค่นับจำนวนแถวของผลลัพธ์เท่านั้น

2.6 การคิวรีข้อมูลด้วยบริการไบโอมาร์ทในโปรแกรมทาเวอร์นา

ไบโอมาร์ทเป็นคลังข้อมูล (Data Warehouse) ที่รวบรวมข้อมูลด้านต่างๆทางชีววิทยาอย่างมากมายและซับซ้อน อนุญาตให้ผู้ที่สนใจเข้าถึงข้อมูลมากมายเหล่านั้นได้ เช่น บริการไบโอมาร์ทของ EBI มีข้อมูลจากฐานข้อมูล *Ensembl, VEGA, DbSNP, UniProt* และ *MSD* [12] และโปรแกรมทาเวอร์นาเอง ก็มีกลไกการคิวรีของไบโอมาร์ทที่สามารถค้นหา และใช้งานข้อมูลเหล่านั้นได้เช่นเดียวกัน

2.6.1. การเพิ่มบริการไบโอมาร์ท

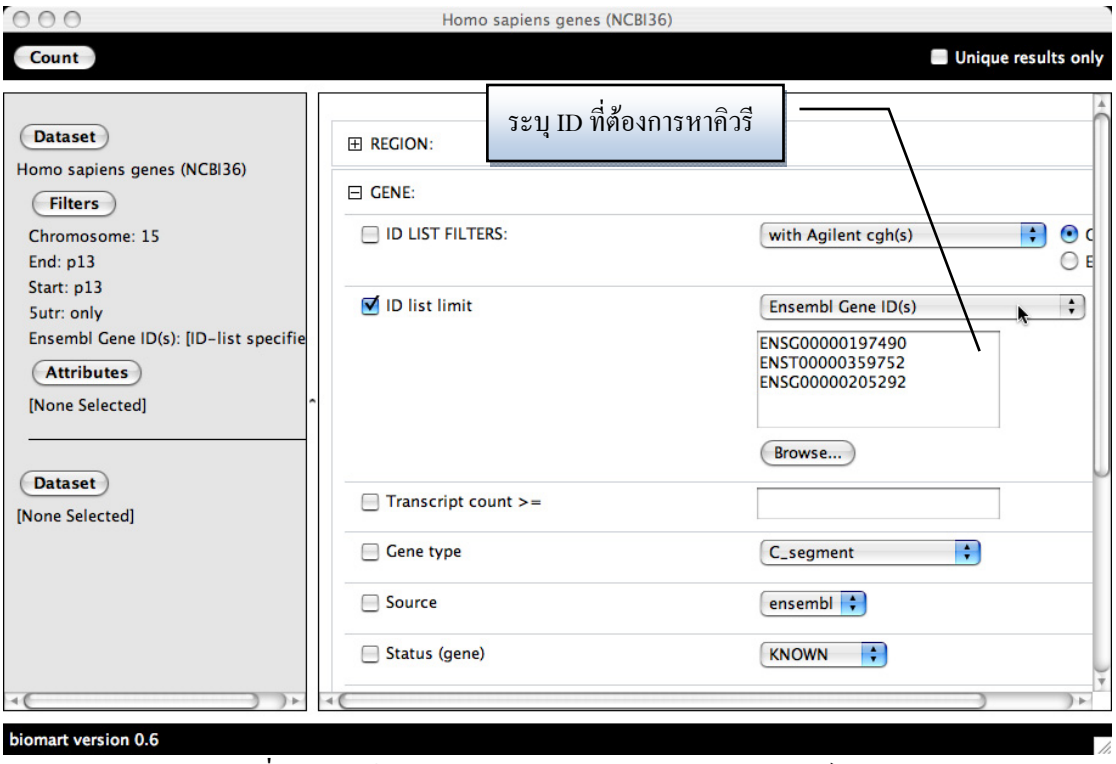
ผู้ใช้สามารถเพิ่มเอนพอยต์ของบริการไบโอมาร์ทใหม่ๆเข้าไปใน *Available service panel* ของโปรแกรมทาเวอร์นา [8][9] หากเอ็นพอยต์สามารถเข้าถึงได้หรือมีการตอบกลับ (Responding) ก็จะแสดงรายชื่อโปรเซสเซอร์หรือบริการต่างๆใน *Available services panel* โดยสามารถแตกโครงสร้างต้นไม้เพื่อเลือกบริการที่ต้องใช้งานได้ ดังรูปที่ 2.27



รูปที่ 2.27 โครงสร้างต้นไม้ของบริการไบโอมาร์ทในโปรแกรมทาเวอร์นา

- **การปรับแต่งฟิลเตอร์ (Configuring Filters)**

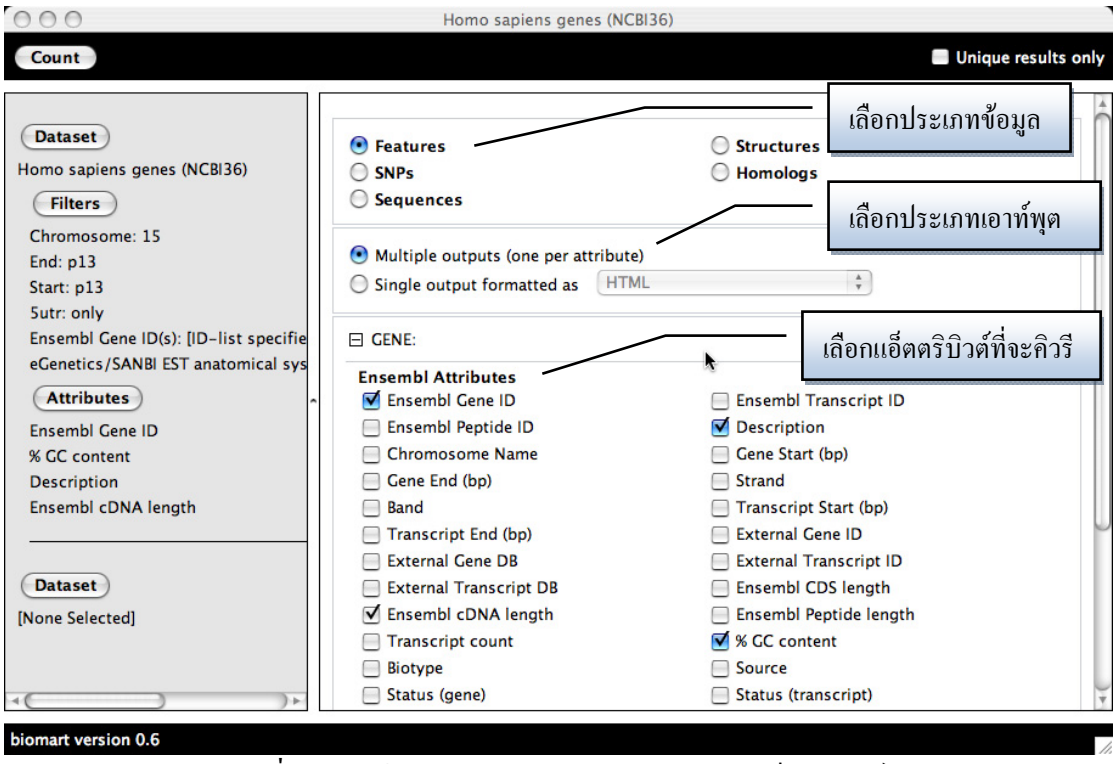
เมื่อเพิ่มบริการไบโอมาร์ทลงใน AME แล้ว สามารถปรับแต่งค่าที่ต่างๆให้ตรงตามความต้องการได้ผ่านหน้าต่าง *'Out of the box'* ดังรูปที่ 2.28 โดยความเร็วในการทำงานของบริการไบโอมาร์ทจะขึ้นอยู่กับความเร็วของเครือข่ายที่ต่อออกสู่อินเตอร์เน็ต เนื่องจากหน้าต่างกราฟิก (GUI) ที่ทำหน้าที่ติดต่อกับผู้ใช้งานของไบโอมาร์ทนั้นต้องดึงข้อมูล (Fetch) จากเครื่องแม่ข่ายที่ให้บริการซึ่งผ่านเครือข่ายอินเตอร์เน็ต ซึ่งการปรับแต่งจะขึ้นอยู่กับความต้องการดึงหรือคิวรีข้อมูลที่ผู้ใช้งานสนใจ โดยฟิลเตอร์จะสามารถปรากฏให้เลือกได้ก็ต่อเมื่อมีการเลือกดาตาเซ็ตแล้วเท่านั้น



รูปที่ 2.28 หน้าต่าง *'Out of the box'* กำหนดค่าฟิลเตอร์ต่างๆ

2.6.2. การปรับแต่งแอ็ตตริบิวต์ (Configuring Attributes)

แอ็ตตริบิวต์หรือฟิลด์ข้อมูลที่ต้องการคิวรีสามารถกำหนด และปรับแต่งที่หน้าจอของแอ็ตตริบิวต์ ซึ่งจะแบ่งออกเป็นหน้าๆตามประเภทของข้อมูลที่สามารถคิวรีได้ คือ *Feature, Structure, SNPs, Homologs* และ *Sequences* ดังแสดงในรูปที่ 2.29 การเลือกแอ็ตตริบิวต์ สามารถเลือกได้เพียงประเภทเดียวเท่านั้นต่อหนึ่งดาตาเซ็ต เนื่องจากแนวคิดการทำงานแบบอะตอมมิก หากผู้ใช้เลือกแอ็ตตริบิวต์ของประเภทข้อมูลอื่นๆ แอ็ตตริบิวต์ที่ถูกเลือกก่อนหน้าก็จะเอาออกโดยอัตโนมัติและแอ็ตตริบิวต์ที่ได้เป็นผลลัพธ์ก็จะขึ้นอยู่กับเงื่อนไขของฟิลเตอร์นั้นๆ

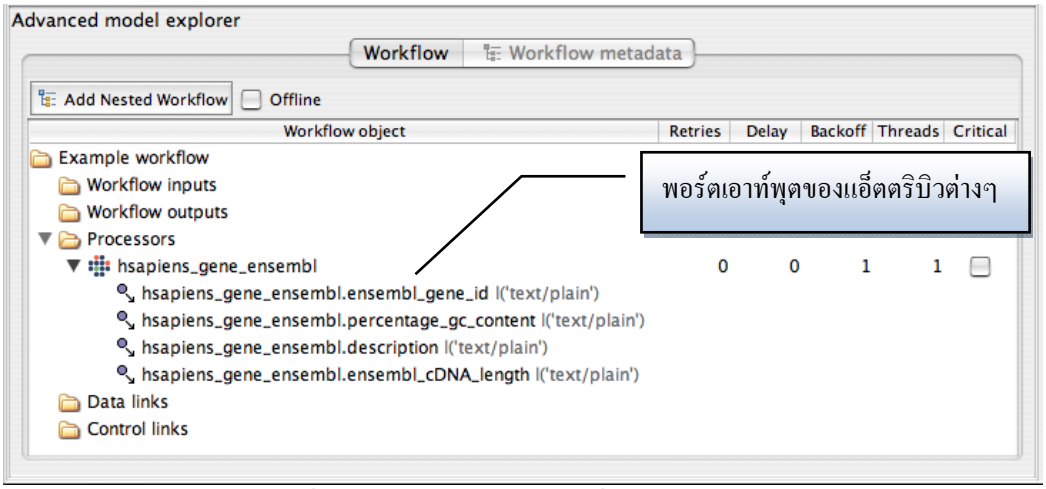


รูปที่ 2.29 หน้าต่าง *'Out of the box'* กำหนดค่าแอ็ตตริบิวต์

2.6.3. โหมดผลลัพธ์ของบริการไบโอมาร์ท (Result modes)

ผลลัพธ์ของการคิวรีด้วยบริการไบโอมาร์ทมีสองโหมดคือ แบบหลายเอาท์พุตและแบบเอาท์พุตเดียวซึ่งขึ้นอยู่กับการเลือกจำนวนของแอ็ตตริบิวต์ ซึ่งการเลือกจำนวนแอ็ตตริบิวต์เพิ่มขึ้นหรือน้อยลง ก็จะทำให้จำนวนเอาท์พุตของบริการแปรผันตามไปด้วย

- หลายเอาท์พุต (Multiple Outputs) - แต่ละแอ็ตตริบิวต์ที่ถูกเลือกจะกลายมาเป็นหนึ่งเอาท์พุตของบริการนั้นๆ ซึ่งจะมีชื่อเพียงพอที่สามารถคาดเดาได้โดยไม่ยากนัก จากรูปที่ 2.30 แสดงบริการใน AME ที่ประกอบด้วยแอ็ตตริบิวต์จำนวน 3 ตัว ซึ่งจะแสดงเป็น 3 เอาท์พุตพอร์ต
- เอาท์พุตเดียว (Single Output) - เช่นเดียวกับแบบหลายเอาท์พุต ขึ้นอยู่กับการเลือกจำนวนแอ็ตตริบิวต์ ในกรณีนี้คือเลือกเพียงหนึ่งแอ็ตตริบิวต์ ก็จะได้เพียงหนึ่งเอาท์พุตปรากฏที่บริการนั้นๆ



รูปที่ 2.30 บริการไบโอมาร์ทที่มีหลายเอาท์พุต

2.7 สรุป

ในบทนี้ได้กล่าวถึงสถาปัตยกรรมของงานวิทยานิพนธ์ และหลักการสำคัญที่เกี่ยวข้องกับเวิร์คโฟลว์, โปรแกรมทาเวอร์นา, เว็บเซอร์วิสและบริการต่างๆที่จะนำมาสร้างเป็นเวิร์คโฟลว์ รวมทั้งรายละเอียดเกี่ยวกับบริการไอโบมาร์ทที่จะเป็นบริการหลักที่เลือกใช้ในวิทยานิพนธ์นี้

การค้นหาบริการไบโอมาร์ทที่มีฟิลเตอร์และแอ็ตตริบิวต์ใดที่ล้าสมัย จะต้องค้นหาโดยวิธีการตรงไปตรงมา (Manual) คือต้องเรียกหน้าต่าง *'Out of the box'* ของบริการไบโอมาร์ทขึ้นมาแล้วสังเกตดูว่าฟิลเตอร์และแอ็ตตริบิวต์ใดที่เลือกไว้ก่อนหน้านั้น ยังปรากฏในหน้าต่าง *'Out of the box'* หรือไม่ หากไม่มีปรากฏแล้ว ก็แสดงว่าฟิลเตอร์และแอ็ตตริบิวต์นั้นๆถูกผู้ให้บริการเอาออกหรือเปลี่ยนไปเป็นชื่อใหม่ เป็นต้น เหล่านี้ล้วนต้องอาศัยประสบการณ์ และการสังเกตจากการใช้บริการไบโอมาร์ทของผู้ใช้งานเอง ยิ่งกว่านั้น จำนวนของบริการไบโอมาร์ทที่อยู่ในเวิร์คโฟลว์, จำนวนแอ็ตตริบิวต์และฟิลเตอร์ที่เรียกใช้ในบริการไบโอมาร์ท, และความซับซ้อนของเวิร์คโฟลว์ ปัจจัยเหล่านี้ ล้วนมีผลต่อการตรวจสอบหรือการค้นหาสาเหตุที่ทำให้เวิร์คโฟลว์ทำงานได้ผลลัพธ์ที่ไม่ตรงกับความต้องการหรือไม่ให้ผลลัพธ์อะไรเลย

ดังนั้นขั้นตอนการตรวจสอบเวิร์คโฟลว์ก่อนการทำงานจึงมีความจำเป็น โดยจะมุ่งเน้นไปที่การตรวจสอบบริการไบโอมาร์ทในเวิร์คโฟวล์สำหรับวิเคราะห์สนิปของมนุษย์ เพราะเป็นบริการหลักที่ใช้สำหรับงานวิจัยและเป็นกรณีศึกษาหลักของวิทยานิพนธ์นี้ ผู้วิจัยจึงได้เสนอ

แนวทางแก้ไขปัญหาโดยการพัฒนาเวิร์คโฟลว์ภายใต้สิ่งแวดล้อมของทาเวอร์นา ที่สามารถตรวจสอบทั้งฟิลเตอร์ และแอ็ตตริบิวต์ของบริการไบโอมาร์ทได้ เพื่อให้ข้อมูลที่มีนัยสำคัญต่อผู้ใช้งานจะได้มีความสะดวกและง่ายต่อการปรับแก้เวิร์คโฟลว์ ซึ่งจะเป็นการประหยัดเวลาในการดีบักลงได้มากดังจะได้กล่าวในบทที่ 5 และบทที่ 6 ต่อไป

บทที่ 3

การเลือกบริการที่เหมาะสมและการสร้างบริการท้องถิ่นในการวิเคราะห์สนิปกุ้ง

3.1 บทนำ

โครงการมายกริดพัฒนาโปรแกรมทาเวอร์นา เพื่อให้ผู้ใช้สามารถเข้าถึงบริการหรือทรัพยากรอันหลากหลายต่างๆที่กระจัดกระจายกันอยู่บนอินเตอร์เน็ต โปรแกรมทาเวอร์นาสามารถเข้าถึงบริการได้กว่า 30,000 ตัวอย่างเช่น ชีวสารสนเทศ วิทยาศาสตร์สุขภาพ ชีวเคมี และดาราศาสตร์ เป็นต้น บริการจำนวนมากเหล่านี้ แต่ละบริการก็มีพารามิเตอร์และวิธีการเข้าถึงที่แตกต่างกันไป โครงการมายกริดออกแบบสถาปัตยกรรมการทำงานของโปรแกรมทาเวอร์นาให้ปกปิดความยุ่งยากซับซ้อนในการเข้าถึงทรัพยากรเหล่านี้ไว้ด้วยการทำงานแบบเวิร์คโฟลว์ ทำให้ผู้ใช้งานเข้าถึงได้โดยง่าย อีกทั้งบริการหรือทรัพยากรจะทำงานอยู่บนระบบที่มีฮาร์ดแวร์ที่ทรงประสิทธิภาพ เช่น ทำงานอยู่บนระบบคลัสเตอร์ขนาดหลายร้อยโหนด เป็นต้น

อย่างไรก็ตามงานวิจัยที่ใช้เทคโนโลยีเวิร์คโฟลว์ ยังจัดว่ามีลักษณะเฉพาะตัว จึงจำเป็นต้องพัฒนาเวิร์คโฟลว์ที่เหมาะสมกับงาน กล่าวคือ การเลือกบริการที่ทำงานได้ถูกต้องตรงกับความต้องการให้มากที่สุด เพราะไม่สามารถที่จะหาบริการที่ใช้ข้อมูลอินพุตและผลิตเอาท์พุตที่ต้องการสำหรับกระบวนการใดๆในงานวิจัยแบบเฉพาะจงเจาะมาใช้งานได้ทันที ซึ่งในบางกรณีอาจต้องมีการปรับแต่งรูปแบบของอินพุตและเอาท์พุตเหล่านั้นบ้างตามความจำเป็น นอกจากนี้ หากไม่สามารถหาบริการที่ต้องการได้ หรือบริการที่หาได้แล้วไม่สามารถตอบสนองความต้องการ เช่น การเกิดปัญหาการเกินเวลาของการส่งผ่านข้อมูลระหว่างบริการ ซึ่งเป็นปัญหาจากเครือข่ายหรือการติดต่อสื่อสารภายนอกโปรแกรมทาเวอร์นา และการไม่คงเส้นคงวาในการทำงานของเวิร์คโฟลว์ที่เรียกใช้บริการต่างๆที่อยู่บนอินเตอร์เน็ต (ทำงานได้บ้างไม่ได้บ้าง) เช่น เวิร์คโฟลว์ไม่ตอบสนองการสั่งงานจากผู้ใช้ซึ่งเป็นปัญหาที่โปรแกรมทาเวอร์นาเอง

ดังนั้นบทนี้จึงนำเสนอการวาดและการเลือกบริการที่เหมาะสม โดยให้ข้อเสนอแนะที่เป็นรูปธรรมและวิธีการใช้เวิร์คโฟลว์ และเสนอบริการท้องถิ่นที่มีความจำเป็นใน

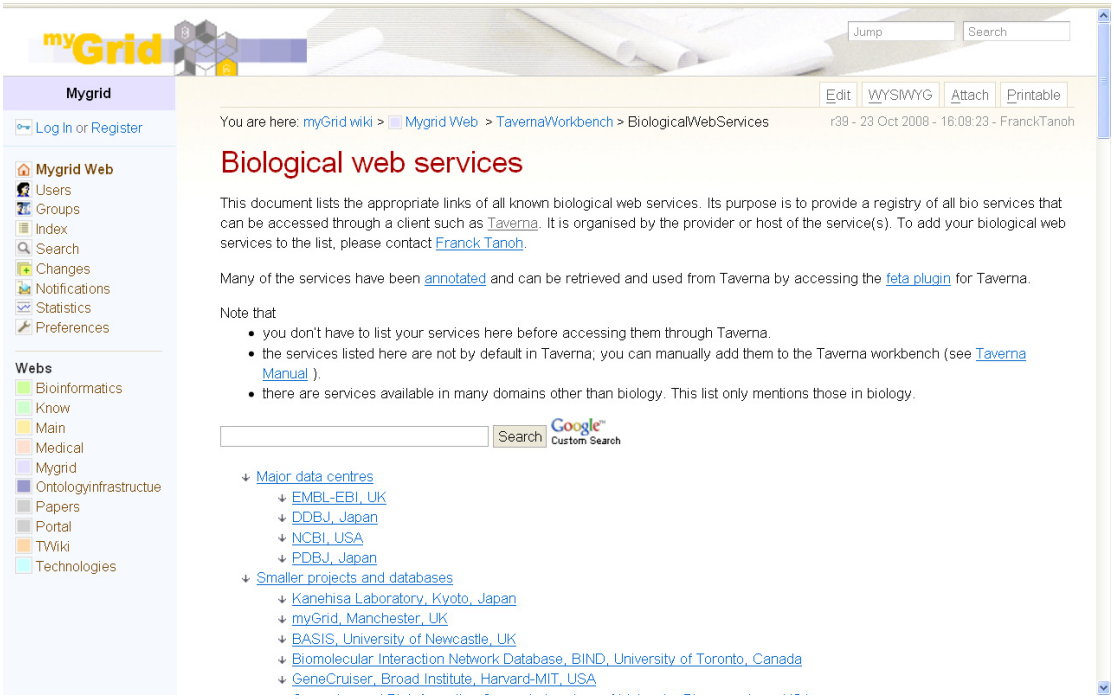
การทำงานสำหรับการแก้ปัญหาในการวิเคราะห์สนิปกุ้ง [38] ที่เป็นงานวิจัยร่วมกับศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์เป็นกรณีศึกษา และอภิปรายผลการทดลอง

3.2 ขั้นตอนการเลือกบริการที่เหมาะสม

การเลือกบริการที่เหมาะสมตรงกับความต้องการในการสร้างเวิร์คโฟลว์ใดๆ ที่เสนอในวิทยานิพนธ์นี้ แบ่งวิธีการค้นหาและการเลือกบริการออกเป็น 2 วิธีคือ การค้นหาบริการจากเว็บพอร์ทอลของมายกริด และการค้นหาบริการจากอินเตอร์เน็ตโดยปลักอินของโปรแกรมทาเวอร์นา ดังนี้

3.2.1 การค้นหาบริการจากเว็บพอร์ทอลของมายกริด

การค้นหาบริการจากเว็บพอร์ทอลของมายกริด ซึ่งรวมไดเรคทอรีของบริการหรือเว็บเซอร์วิสต่างๆที่โปรแกรมทาเวอร์นาสามารถเข้าถึงได้ เรียกว่า *Biological Web Service*s ดังรูปที่ 3.1 โดยมี URL คือ http://www.mygrid.org.uk/wiki/Mygrid/BiologicalWebServices ผู้ใช้สามารถเชื่อมต่อไปยังเว็บไซต์ที่ของผู้ให้บริการ และสามารถทดลองใช้บริการเหล่านั้นกับเว็บเบราเซอร์ได้ทันที เช่น บริการ EBI หรือไบโอมาร์ทเป็นต้น



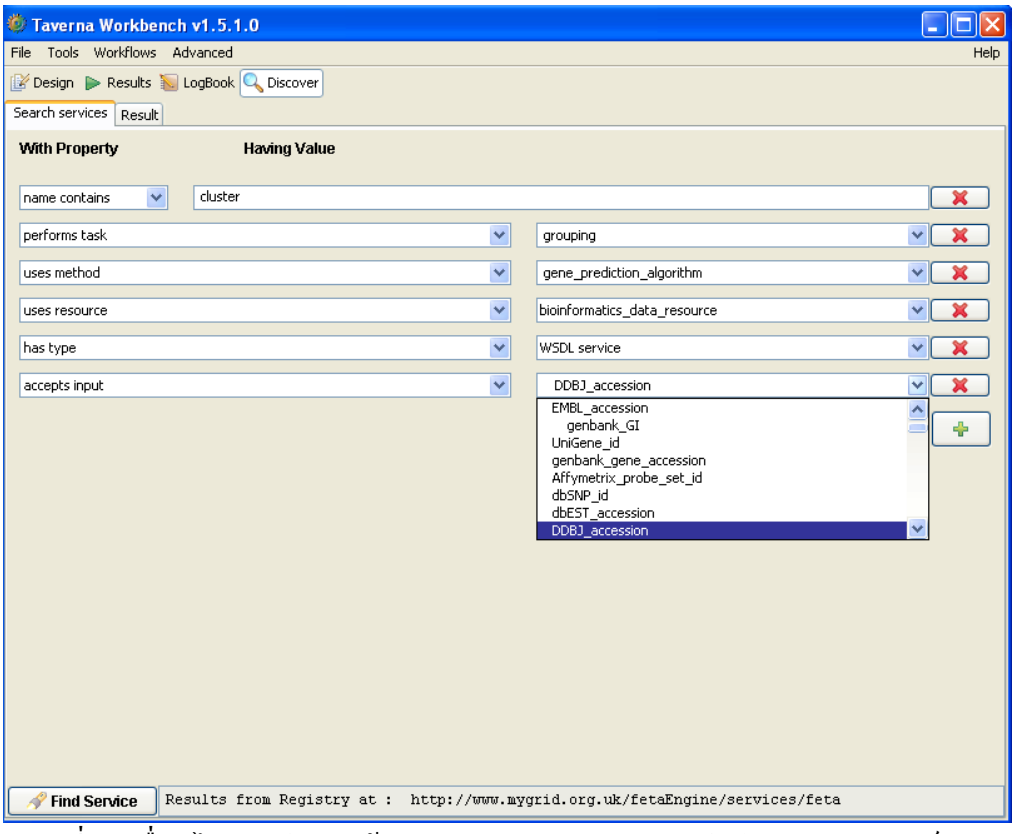
รูปที่ 3.1 เว็บไซต์ Biological web services ของมายกริด

3.2.2 การค้นหาบริการจากอินเตอร์เน็ตโดยปลักอินของโปรแกรมทาเวอร์นา

โปรแกรมทาเวอร์นามีปลักอินชื่อ *feta* เป็นตัวช่วยค้นหาบริการหรือเว็บเซอร์วิสต่างๆที่กระจัดกระจายบนเครือข่ายอินเตอร์เน็ต *feta* ค้นหาเว็บเซอร์วิสโดยใช้พื้นฐานจากคำอธิบายของหน้าที่ในการทำงานของบริการนั้นๆ (Description) ด้วย Service ontology ซึ่งจะเก็บอยู่ที่รีจิสทรีของบริการหรือ UDDI ของมายกริด หากพบบริการที่น่าจะเหมาะสมตรงตามความต้องการก็สามารถทดลองการใช้งานง่ายๆได้ ซึ่งอาจจะพบว่า ในงานหนึ่งๆนั้น มีบริการที่ทำงานคล้ายคลึงกันเป็นจำนวนมาก เนื่องจากการบริการเหล่านั้นได้รับการออกแบบโดยคำนึงถึงแนวคิดการทำงานแบบอะตอมมิกหรือไม่อาจจะแบ่งย่อยการทำงานลงได้อีก ซึ่งจะเป็นประโยชน์ต่อการดีบักหรือตรวจสอบการทำงาน จากรูปที่ 3.2 เงื่อนไขในการค้นหาของ *feta* สามารถกำหนดได้หลายอย่างดังนี้ [8][9]

- ค้นหาบริการที่รับอินพุตในรูปแบบข้อมูลที่ต้องการ เช่น หมายเลขข้อมูลของฐานข้อมูล DDBJ

- ค้นหาบริการที่ผลิตเอาท์พุตในรูปแบบข้อมูลที่ต้องการ เช่น หมายเลขไอดีของสนิป
- ค้นหาบริการที่ทำหน้าที่ใดหน้าที่หนึ่งที่ต้องการ เช่น Grouping
- ค้นหาบริการที่มีอัลกอริทึมหรือ Method ที่ต้องการ เช่น Gene prediction algorithm หรืออัลกอริทึมในการทำนายยีน
- ค้นหาบริการที่มีฟังก์ชันการทำงานบางอย่างของโปรแกรมประยุกต์ เช่น Secondary protein structure prediction หรือการทำนายโครงสร้างขั้นที่สองของโปรตีน
- ค้นหาชนิดของบริการที่ต้องการเช่น บริการ WSDL หรือบริการ Soaplab
- ค้นหาบริการจากชื่อและคำอธิบายการทำงานของบริการ เช่น Sapiens (มนุษย์)



รูปที่ 3.2 เงื่อนไขต่างๆในการค้นหาบริการของปลักอิน Feta ในโปรแกรมทาเวอร์นา

จากเงื่อนไขต่างๆของการค้นหาโดย *Feta* จะเห็นว่า การระบุเงื่อนไขจะสามารถค้นหาบริการที่ตรงกับการทำงานของผู้ใช้ได้ เช่น ในวิทยานิพนธ์นี้ ระบุเงื่อนไขการค้นหาบริการชนิด Soaplab และระบุฟังก์ชันการทำงานว่า Translation หรือการถอดรหัสจากลำดับนิวคลีโอไทด์เป็นลำดับกรดอะมิโน เป็นต้น ในการใช้งานจริงนั้น ผู้ใช้อาจจะระบุชื่อหรือคำอธิบายการทำงานสั้นๆง่ายๆ และระบุประเภทของบริการ ก็เพียงพอที่จะหาบริการที่ตรงกับความต้องการจากรายการบริการต่างๆที่ *Feta* นำเสนอได้

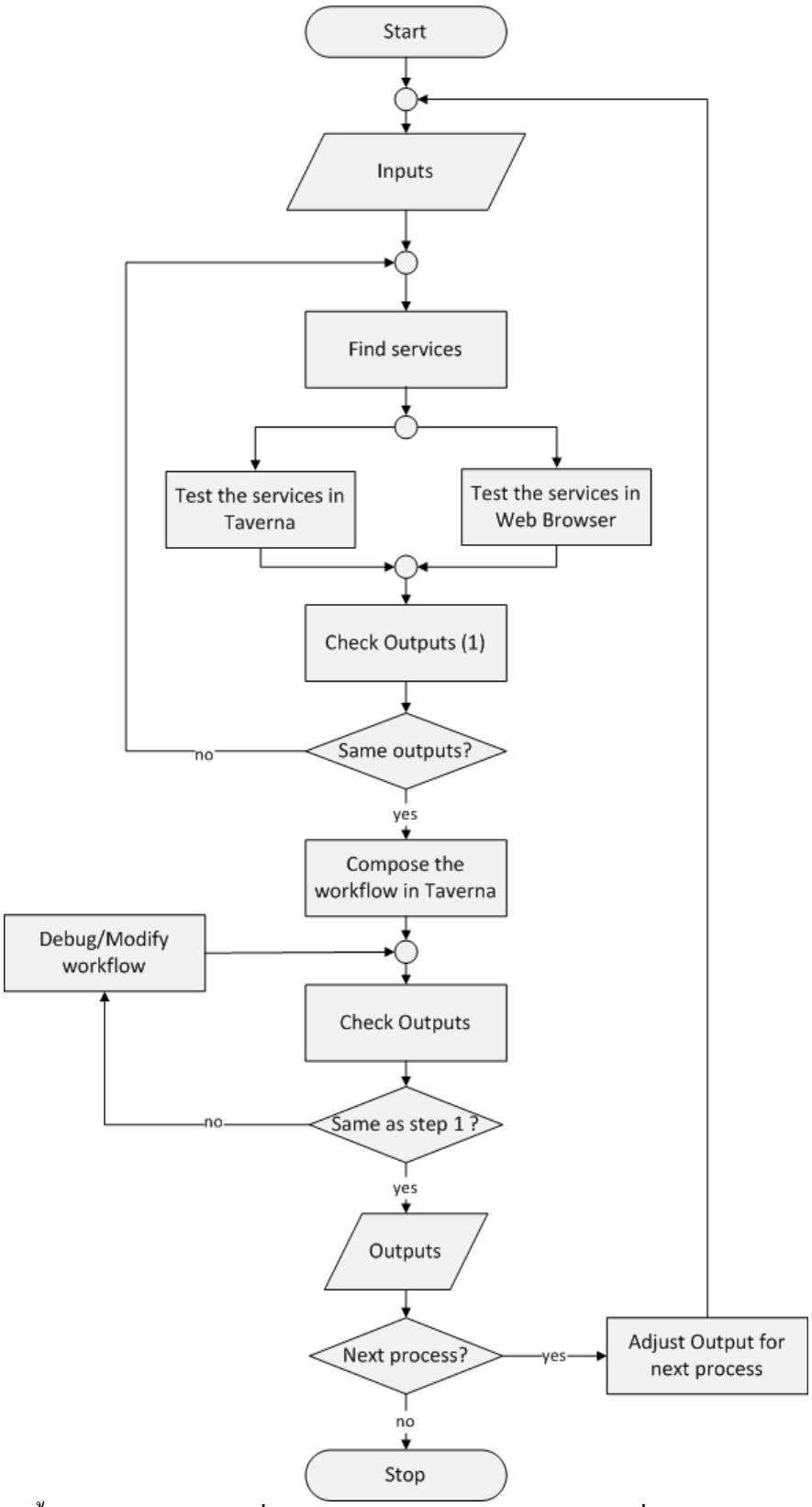
3.3 ขั้นตอนการพัฒนาเวิร์คโฟลว์

ขั้นตอนการพัฒนาเวิร์คโฟลว์ที่จะกล่าวในส่วนนี้ คือการออกแบบ Flowchart หรือ Data flow Diagram ของระบบงานวิจัยเสร็จสิ้นแล้ว ผู้วิจัยมองเห็นอินพุต การไหลของข้อมูลและเอาท์พุตโดยรวมจากการออกแบบได้ กระบวนการต่อไปคือ ขั้นตอนของการพัฒนาเวิร์คโฟลว์ของระบบงานตามที่ได้ออกแบบไว้แล้วดังรูปที่ 3.3

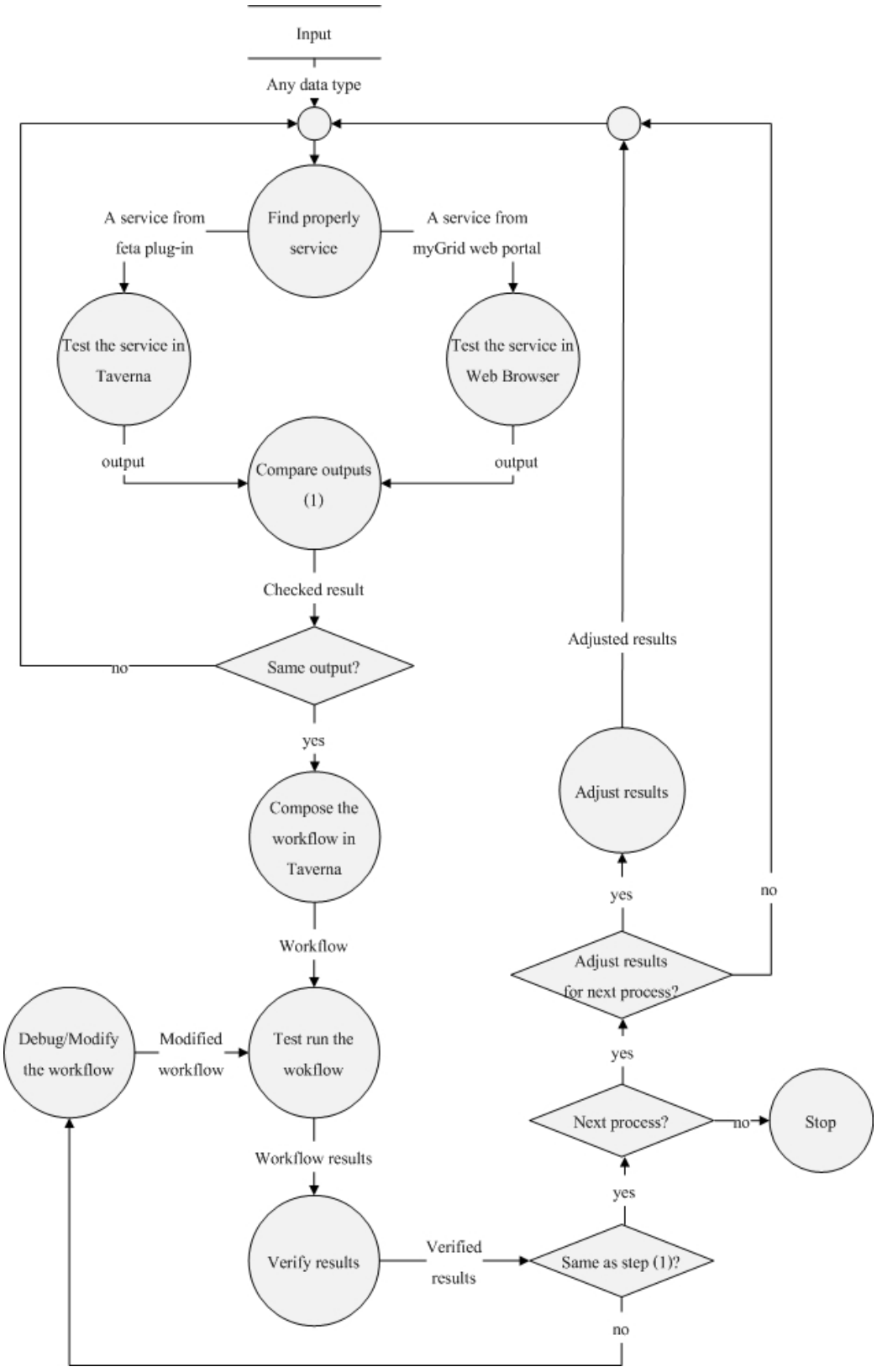
จากรูปที่ 3.3 สามารถอธิบายข้อมูลที่ไหลในกระบวนการพัฒนาเวิร์คโฟลว์ได้ด้วย Data flow diagram ดังรูปที่ 3.4 โดยขั้นเริ่มต้น มีข้อมูลอินพุตของงานวิจัยจากการออกแบบ Flowchart หรือ Data flow aiagram แล้ว จากนั้นเริ่มค้นหาบริการที่สามารถรองรับรูปแบบของอินพุตได้ หรืออาจจะปรับแต่งรูปแบบของอินพุตเพียงเล็กน้อย โดยไม่ใช้เวลาหรือแรงงานมากเกินความจำเป็น ดังที่ได้กล่าวในหัวข้อที่ 3.2 ก็สามารถนำอินพุตที่ปรับแต่งนี้ไปใช้งานกับบริการที่เลือกมาได้

- ทดสอบการทำงานของบริการที่เลือกมานี้กับอินพุต ทั้งในโปรแกรมทาเวอร์-นา (ไม่ต้องวาดเป็นเวิร์คโฟลว์ก่อน) และเว็บเบราเซอร์ แล้วนำผลลัพธ์การทำงานมาเปรียบเทียบกันว่า ถูกต้องตรงกันหรือไม่ เพื่อตรวจสอบความถูกต้องหรือดีบักซึ่งกันและกัน และเป็นการเพิ่มความเชื่อมั่นในการทำงานของบริการที่นำมาประกอบเป็นเวิร์คโฟลว์ในขั้นตอนต่อไป
- หากผลการเปรียบเทียบไม่ถูกต้องตรงกัน แสดงว่าบริการที่เลือกมานั้นยังไม่อาจจะนำมาใช้งานได้ เพราะหากนำมาใช้แล้ว ในอนาคตบริการมีการเปลี่ยนแปลงก็จะทำให้ยากต่อการดีบักหรือตรวจสอบ

- หากผลการเปรียบเทียบถูกต้องตรงกัน ขั้นตอนต่อไปคือนำบริการนี้ไปทดสอบสร้างเป็นเวิร์คโฟลว์ในโปรแกรมทาเวอร์นา แล้วทดสอบการทำงานและตรวจสอบผลลัพธ์การทำงานว่า ถูกต้องเหมือนกับที่ใช้งานกับเว็บเบราเซอร์หรือไม่
- หากผลลัพธ์การทำงานไม่ถูกต้อง แสดงว่ามีความผิดพลาดจากการสร้างเวิร์ค-โฟลว์หรือสาเหตุใดๆก็ได้ ผู้ใช้งานอาจจะตรวจสอบหรือปรับแต่ง Data flow ในเวิร์คโฟลว์อย่างละเอียดถี่ถ้วน ซึ่งโปรแกรมทาเวอร์เองก็มีเบรกพอยต์และรายงานสถานะของโปรเซสเป็นเครื่องมือดีบัก และลองทดสอบการทำงานซ้ำอีกครั้งหนึ่ง
- หากผลการทำงานจากเวิร์คโฟลว์ถูกต้องแล้ว ก็พิจารณาต่อว่าเอาท์พุตที่ได้มานี้ จะต้องไปเป็นอินพุตของกระบวนการหรือบริการถัดไปตามที่ได้ออกแบบไว้หรือไม่ ก็เริ่มต้นหาบริการที่เหมาะสมกับอินพุตนี้ต่อไป
- ทำตามกระบวนการดังกล่าวไปเรื่อยๆ จนกระทั้งครบทุกกระบวนการก็จะสามารถหาบริการที่ผลิตผลลัพธ์ที่ถูกต้องเหมาะสมกับงานได้



รูปที่ 3.3 ขั้นตอนการหาบริการที่เหมาะสมและการพัฒนาเวิร์คโฟลว์ที่เสนอในวิทยานิพนธ์นี้



รูปที่ 3.4 Data flow diagram อธิบายข้อมูลที่ไหลในกระบวนการพัฒนาเวิร์คโฟลว์

3.4 การออกแบบเวิร์คโฟลว์และเว็บเซอร์วิสสำหรับการวิเคราะห์สนิปกุ้ง

การสร้างเวิร์คโฟลว์เพื่อวิเคราะห์สนิปกุ้ง มีประโยชน์ในการพัฒนาสายพันธุ์ เป็นความร่วมมือกับสถานวิจัยจีโนมิกส์และชีวสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาสงขลานครินทร์ ซึ่งเป็นกรณีศึกษาเบื้องต้นก่อนที่จะนำระบบไปปรับใช้งานกับกรณีศึกษาหลักต่อไป

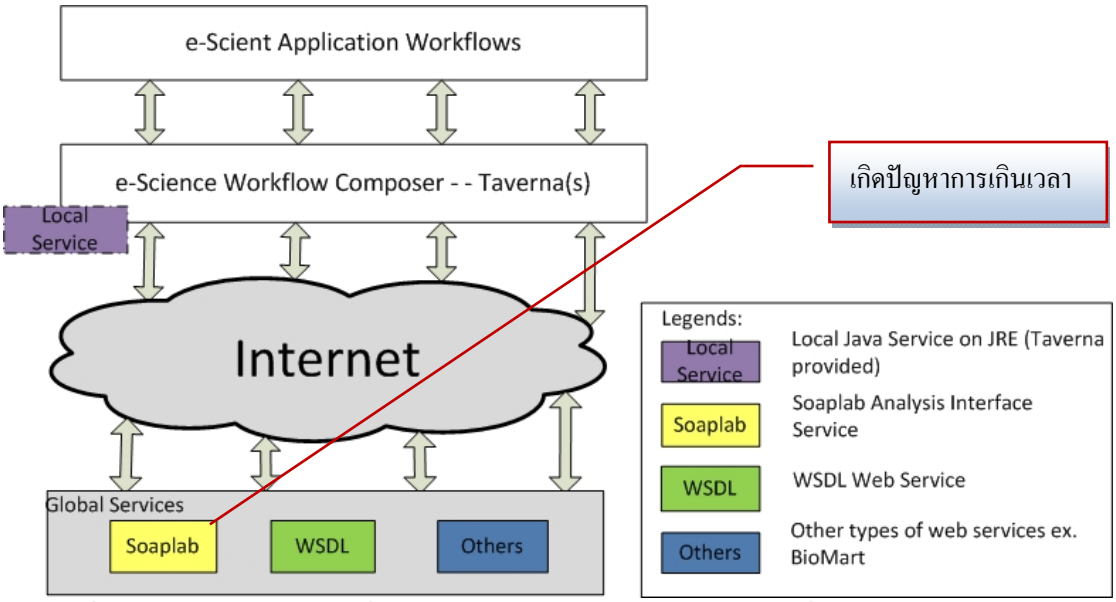
เวิร์คโฟลว์จะประกอบด้วยเว็บเซอร์วิสต่างๆ ที่ทำงานประสานกันไปตามหน้าที่ของเว็บเซอร์วิสนั้นๆ โดยการพัฒนาจะเริ่มตั้งแต่การค้นหาเว็บเซอร์วิสที่จะนำมาใช้งาน ซึ่งเลือกใช้ Emboss web service ของสถาบัน EBI [37] เป็นหลัก เนื่องจากโปรแกรมทาเวอร์นาสามารถเข้าถึงได้อยู่แล้ว และมีการใช้งานที่ไม่ยุ่งยากซับซ้อน สามารถทดสอบความถูกต้องการทำงานของบริการโดยการใช้เว็บเบราเซอร์ได้ ทำให้ง่ายต่อการตรวจสอบความถูกต้องของแต่ละเว็บเซอร์วิส ซึ่งเป็นไปตามกระบวนการที่นำเสนอในการเลือกบริการที่เหมาะสม

นอกจากนี้ ยังพัฒนาเว็บเซอร์วิสท้องถิ่น (Local web services) ในการทำงานด้านการสร้างโครงสร้างต้นไม้สายวิวัฒนาการ (Phylogenetics Tree) ขึ้นมาใช้งานเอง โดยใช้ Soaplab Analysis Tool เป็นเครื่องมือช่วยสร้างตัวเชื่อมประสานหรือ Web service interface ให้กับโปรแกรมประยุกต์หรือแอ็พพลิเคชันต่างๆ ด้านการสร้างโครงสร้างต้นไม้สายวิวัฒนาการเพื่อให้โปรแกรมทาเวอร์นาเข้าถึงได้ เนื่องจากการใช้งานเว็บเซอร์วิสของสถาบัน EBI ในกระบวนการนี้ มีการวิเคราะห์และถ่ายโอนข้อมูลจำนวนมากในแต่ละบริการ ทำให้มีปัญหาการเกินเวบาเกิดขึ้นและความไม่คงที่ในการทำงานส่งผลให้เวิร์คโฟลว์ทำงานได้ไม่สำเร็จ และจำเป็นต้องสร้างเว็บเซอร์วิสท้องถิ่นขึ้นมาใช้งานเอง ซึ่งในบทนี้จะแสดงรายละเอียดต่างๆเกี่ยวกับกรณีศึกษาและเบื้องหลังของงาน วิธีการทำงานแบบเดิมและแบบใหม่ บริการหรือเว็บเซอร์วิสที่เลือกใช้งานในการพัฒนาเวิร์คโฟลว์ การแก้ไขปัญหาการเกินเวลาและความไม่คงเส้นคงวาในการทำงาน การพัฒนาบริการท้องถิ่นและผลการทดลอง ดังต่อไปนี้

3.4.1 กรณีศึกษาและเบื้องหลังของงาน

การวิจัยมุ่งเน้นการศึกษายีนชื่อ *Amylase* ของกุ้งแชบ๊วย เพราะว่ามีรูปแบบที่ดีในการศึกษาด้านกระบวนการปรับตัว (Adaptation) และวิวัฒนาการ ตัวอย่างของ *Penaeus merguiensis* (กุ้ง) เก็บจากแหล่งอาศัย 3 แหล่งในอ่าวไทย คือ ตราด (TDE), สงขลา (SKE), และ

สุราษฎร์ธานี (SRE) [13] ซึ่งสนิปอาจจะเป็นสิ่งสำคัญต่อการศึกษาในอนาคตด้านการเติบโตและการทนต่อโรคของกุ้ง หลังจากที่หาบริการที่ต้องการใช้งานครบทุกบริการ ผู้วิจัยได้พัฒนาได้พัฒนาเวิร์คโฟลว์และแต่พบว่าเวิร์คโฟลว์ไม่สามารถทำงานได้สำเร็จเนื่องจาก Data flow ระหว่างเว็บเซอร์วิสเกิดการเกินเวลาในกระบวนการสร้างโครงสร้างต้นไม้สายวิวัฒนาการดังรูปที่ 3.5 ปัญหานี้เกิดเมื่อกระบวนการใช้ข้อมูลในการหาความน่าจะเป็นมากกว่า 2,000 ชุด ซึ่งเป็นปัญหาจากฝั่งผู้ให้บริการเว็บเซอร์วิสเอง ในที่นี่คือ EBI แม้ว่าจะมีความเป็นไปได้ที่จะทำงานดังกล่าวแบบคัดลอกและวาง แต่เนื่องจากข้อมูลมีจำนวนมาก ทำให้ยากในการทำงานแบบคัดลอกและวางดังกล่าว อีกทั้งยังต้องใช้เวลามากอีกด้วย



รูปที่ 3.5 ภาพรวมการติดต่อสื่อสารของโปรแกรมทาเวอร์นาและจุดที่เกิดปัญหาการเกินเวลา

วิธีการแก้ไขปัญหาที่วิทยานิพนธ์นี้นำเสนอคือ การสร้างบริการท้องถิ่นที่สามารถติดต่อสื่อสารกับเวิร์คโฟลว์ได้ โดยจะเน้นไปที่กระบวนการสร้างโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์ (Nucleotide) และกรดอะมิโน (Amino Acid) ซึ่งตัวช่วยที่สามารถติดต่อสื่อสารนี้เรียกว่า Web service interfaces โดยจะทำหน้าที่เชื่อมประสานให้กับแต่ละกระบวนการของงานในการสร้างโครงสร้างต้นไม้สายวิวัฒนาการในเวิร์คโฟลว์ ซึ่งสามารถเข้าถึงได้โดยโปรแกรมทาเวอร์นา ตัวเชื่อมประสานเว็บเซอร์วิสในวิทยานิพนธ์ใช้โปรแกรม Soaplab Analysis Tool เป็นเครื่องมือในการพัฒนา

เวิร์คโฟลว์ที่พัฒนาในวิทยานิพนธ์ประกอบด้วยระบบงานดังนี้คือ BLAST (Basic Local Alignment and Search Tool) [41], การเรียงลำดับแบบหลายลำดับของลำดับนิวคลีโอไทด์และกรดอะมิโน (Multiple sequence alignment), การทำนายโครงสร้างขั้นที่สองของโปรตีน (Prediction of the protein secondary structure) และการทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์และกรดอะมิโน (Prediction of the phylogenetic tree) ผู้วิจัยใช้โปรแกรมทาเวอร์นาในการรวบรวมเว็บเซอร์วิส ที่เกี่ยวข้องกับระบบงานเข้ามาวาดเป็นเวิร์คโฟลว์ได้แก่ บริการ Web API for Biology จาก DDBJ [40], บริการ EBI [37], บริการท้องถิ่น Java และบริการท้องถิ่นที่สร้างขึ้นมาเอง ความถูกต้องของผลการทำงานในแต่ละบริการตรวจสอบโดยศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์

3.4.2 วิธีการทำงานแบบเดิม (Copying and Pasting)

การวิเคราะห์สนิปในวิธีการทำงานแบบเดิมหรือการคัดลอกแล้ววาง (Copying and Pasting) เป็นขั้นตอนการทำงานด้วยมือโดยการคัดลอกและวางสลับกันระหว่างแอ็พพลิเคชันทางด้านชีวสารสนเทศศาสตร์ ซึ่งทำงานแบบออฟไลน์กับเว็บแอ็พพลิเคชันที่ทำงานบนอินเตอร์เน็ต การวิเคราะห์สนิปแบบดั้งเดิมนี้ประกอบด้วยแอ็พพลิเคชัน และเว็บแอ็พพิลเคชันออนไลน์ต่างๆ ดังต่อไปนี้ [14]

1. โปรแกรม BLAST กับลำดับนิวคลีโอไทด์ เป็นการค้นหาข้อมูลของลำดับนิวคลีโอไทด์ของกุ้งที่ได้จากสกัดจากห้องปฏิบัติการด้วยโปรแกรม BLAST ที่เว็บไซต์ของ NCBI เพื่อหาว่าลำดับนิวคลีโอไทด์ที่สนใจเป็นยีน *Amylase* หรือไม่

2. โปรแกรม Clustal X และ GENEDOC กับลำดับนิวคลีโอไทด์ เป็นการทำ Multiple alignment หรือการเรียงลำดับแบบหลายลำดับของลำดับนิวคลีโอไทด์โดยใช้โปรแกรม Clustalx และ Genedoc ที่เครื่องคอมพิวเตอร์ท้องถิ่นส่วนบุคคลเพื่อดูว่าลำดับต่างๆที่นำมาวิเคราะห์มีส่วนคล้ายคลึงกันมากน้อยเพียงใด

3. โปรแกรม Protein sequence analysis กับลำดับนิวคลีโอไทด์ เป็นการ Translation หรือถอดรหัสจากลำดับนิวคลีโอไทด์ซึ่งเป็น DNA ไปสู่ลำดับกรดอะมิโนโดยใช้โปรแกรม Protein Sequence Analysis (PSA) ที่เครื่องคอมพิวเตอร์ท้องถิ่นส่วนบุคคลแล้วนำลำดับอะมิโนที่ได้ไปสู่กระบวนการวิเคราะห์ต่อไป

4. โปรแกรม Clustal X และ GENEDOC กับลำดับกรดอะมิโน เป็นการทำ Multiple alignment การเรียงลำดับแบบหลายลำดับของลำดับกรดอะมิโนที่ได้จากขั้นตอนที่ 3 โดยใช้โปรแกรม Clustalx และ Genedoc ที่เครื่องคอมพิวเตอร์ท้องถิ่นส่วนบุคคลเพื่อดูว่าลำดับต่างๆที่นำมาวิเคราะห์มีส่วนคล้ายคลึงกันมากน้อยเพียงใด

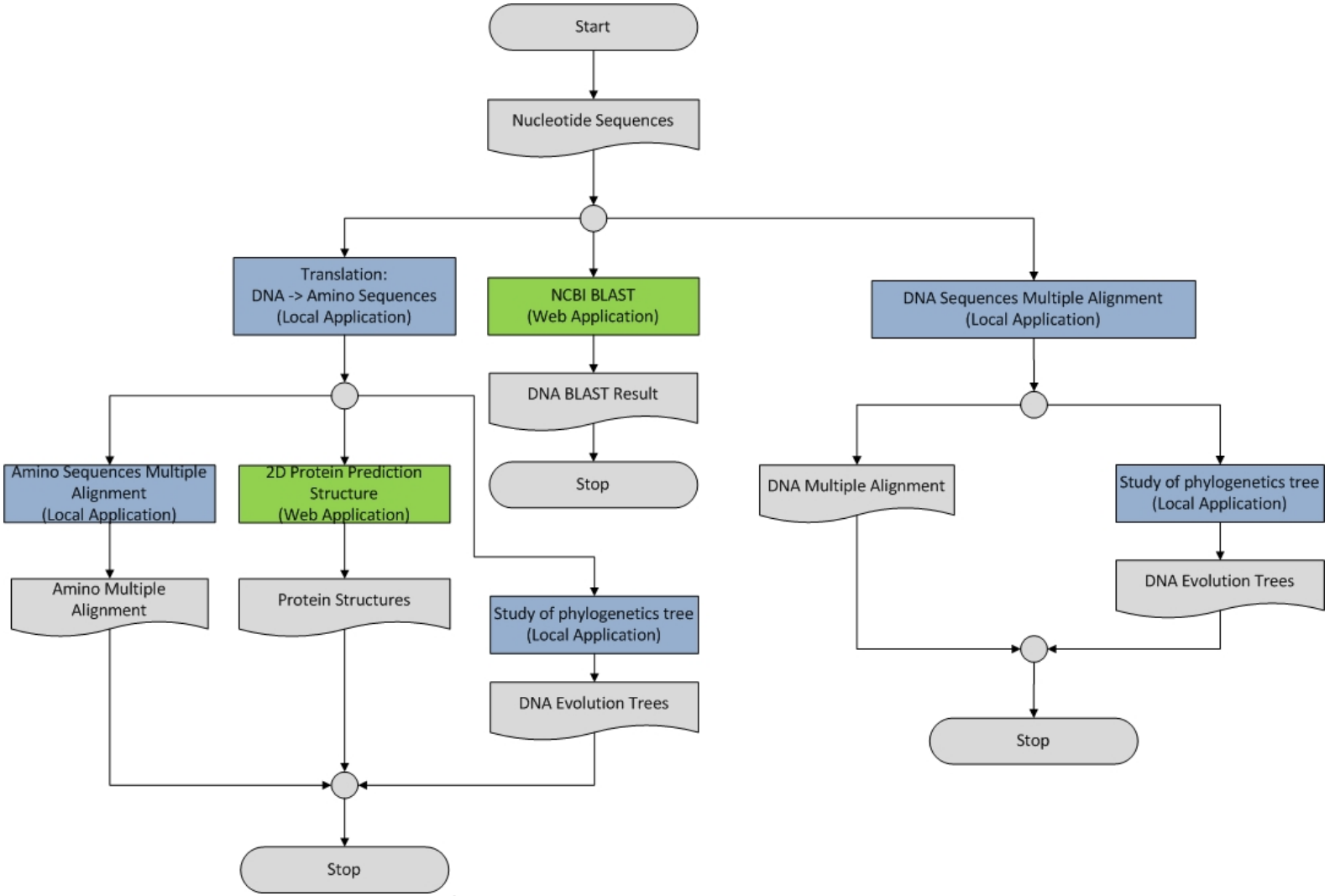
5. โปรแกรม BioEdit กับลำดับนิวคลีโอไทด์และลำดับกรดอะมิโน เป็นการศึกษาของความสัมพันธุ์ทางวิวัฒนาการของสายพันธุ์ โดยวิเคราะห์จากโครงสร้างต้นไม้หรือ Phylogenetics tree ซึ่งสร้างได้จากโปรแกรม BioEdit ที่เครื่องคอมพิวเตอร์ท้องถิ่นส่วนบุคคลของนักวิจัยเอง โดยจะวิเคราะห์ทั้งลำดับนิวคลีโอไทด์และลำดับกรดอะมิโน

6. เว็บแอ็พพลิเคชันชื่อโปรแกรม PSIPRED ที่ http://bioinf.cs.ucl.ac.uk/psipred/ กับลำดับกรดอะมิโน เป็นการทำนายโครงสร้างขั้นที่สองของโปรตีน (Protein secondary structure) โดยใช้ข้อมูลจากลำดับกรดอะมิโนซึ่งมีเว็บแอพพลิชันที่ให้บริการคือ PSIRRED

จากขั้นตอนของการทำงานทั้ง 1-6 ขั้นตอน สามารถสรุปความสัมพันธ์ของแต่ละกระบวนการได้รูปที่ 3.6 ซึ่งแสดงการใช้งานแอ็พพลิเคชันบนเว็บเพจด้วยบล็อกสีเขียว และการใช้งานแอ็พพลิเคชันที่เครื่องคอมพิวเตอร์ท้องถิ่นส่วนบุคคลด้วยบล็อกสีน้ำเงิน

3.4.3 บริการหรือเว็บเซอร์วิสที่เลือกใช้งานและการพัฒนาเวิร์คโฟลว์

รูปที่ 3.6 แสดงภาพรวมของการทำงาน และเห็นลักษณะการไหลของข้อมูลในกระบวนการวิเคราะห์สนิป โดยมีข้อมูลที่เป็นอินพุตให้กับระบบก็คือลำดับนิวคลีโอไทด์ของกุ้งที่ได้จากห้องปฏิบัติการ ในขั้นตอนนี้สามารถเริ่มหาบริการที่เหมาะสมกับความต้องการมาใช้งานตามกระบวนการในรูปที่ 3.3 และพัฒนาเวิร์คโฟลว์ด้วยโปรแกรมทาเวอร์นา ในการพัฒนาเวิร์คโฟลว์นี้ได้ใช้บริการท้องถิ่นของ Java ในการจัดการ Data flow ระหว่างบริการต่างๆด้วย ในที่นี้คือ *'Split string into string list'* (บล็อกสีม่วงในรูปที่ 3.7) ทำหน้าที่ส่งลำดับเบสไปเข้าไปทำงานกับโปรแกรม BLAST ทีละลำดับดังแสดงในตารางที่ 3.1



รูปที่ 3.6 ลำดับของกระบวนการวิเคราะห์สนิปของกุ้ง

ตารางที่ 3.1 แสดงตัวอย่างการทำงานของบริการ *'Split string into string list'*

Input Type	String
Input Example	'ATCGATCGATCGATCGATCG\n ATCGATCGATCGATCGATCG\n ATCGATCGATCGATCGATCG\n'
Output Type	String list
Output Example	'ATCGATCGATCGATCGATCG' 'ATCGATCGATCGATCGATCG' 'ATCGATCGATCGATCGATCG'
Expression	'\n'

บริการหรือเว็บเซอร์วิสหลักที่เลือกมาใช้งานเป็นบริการของชุดโปรแกรม Emboss จากสถาบัน EBI โดยมีบริการที่ทำงานได้ตรงตามความต้องการและเป็นที่รู้จักกันอย่างแพร่หลาย ตารางที่ 3.2 แสดงรายชื่อบริการของชุดโปรแกรม Emboss ที่เกี่ยวข้องและตรงตามความต้องการของกระบวนการทำงาน (บล็อกสีเหลืองในรูปที่ 3.7) กระบวนการทำงานของโปรแกรม BLAST เรียกใช้บริการจาก DDBJ (บล็อกสีเขียวในรูปที่ 3.7) ซึ่งสามารถระบุได้ว่า จะค้นหาลำดับนิวคลีโอไทด์ในฐานข้อมูลอะไรบ้าง ในงานวิจัยนี้เลือกใช้ฐานข้อมูล DDBJNEW [14] โปรแกรม BLAST ของ DDBJ สามารถเข้าถึงได้ด้วยโปรแกรมทาเวอร์นาและเว็บเบราเซอร์ นอกจากนี้ยังสามารถทดลองการใช้งานจากเว็บเบราเซอร์ได้เช่นเดียวกันกับบริการของ EBI เวิร์คโฟลว์ที่พัฒนาเสร็จแล้วแสดงดังรูปที่ 3.7 โดยมีพารามิเตอร์ของแต่บริการหรือเว็บเซอร์วิสที่กำหนดไว้ในเวิร์คโฟลว์ดังตารางที่ 3.3

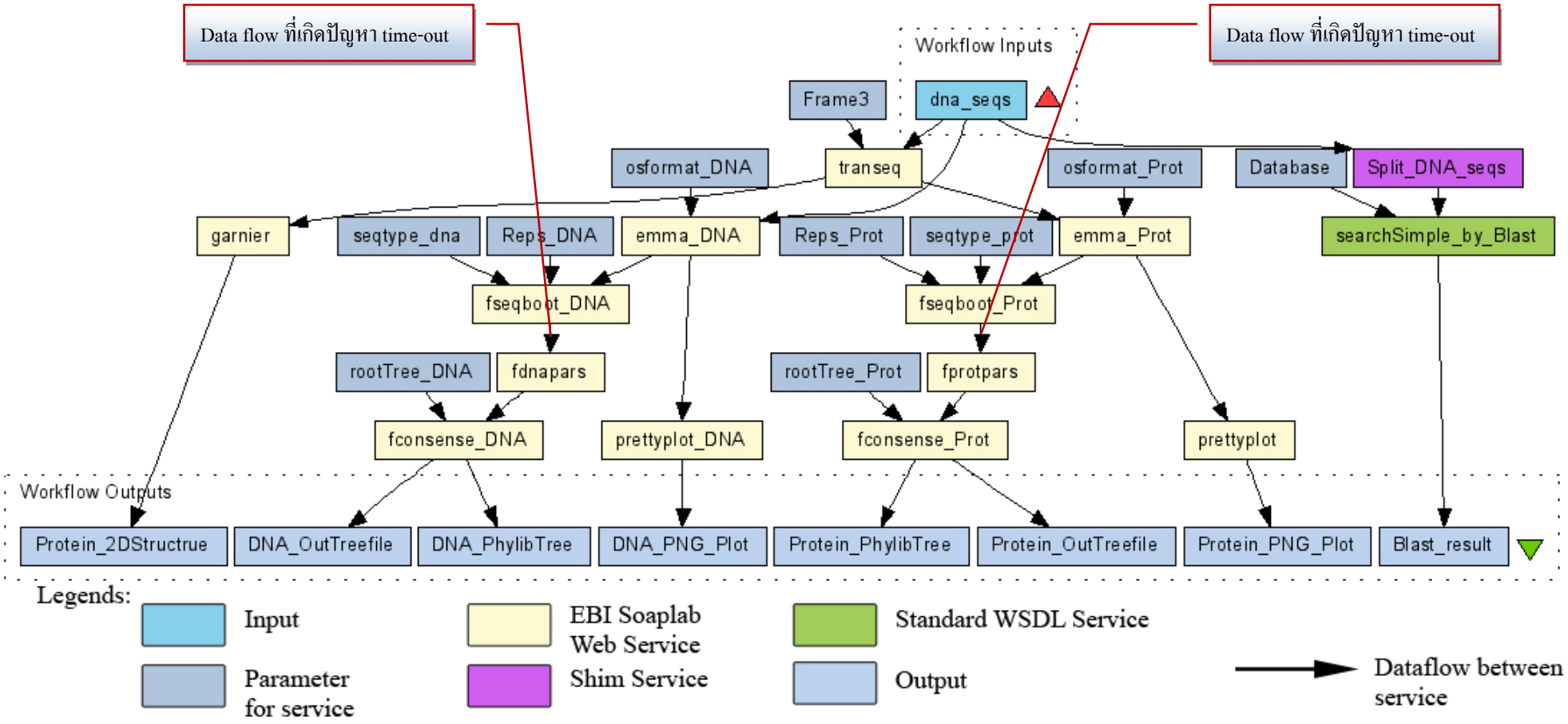
ในกระบวนการทำนาย หรือการสร้างโครงสร้างต้นไม้สายวิวัฒนาการของสิ่งมีชีวิตใดๆ (ในวิทยานิพนธ์นี้คือกุ้งแชบ๊วย) จะใช้บริการ *fseqboot* หรืออัลกอริทึมในการทำ Bootstrapping replicate กับลำดับข้อมูล *fseqboot* จะอ่านชุดข้อมูลและสร้างชุดข้อมูลของลำดับนิวคลีโอไทด์หรือกรดอะมิโนหลายๆชุดตามที่ผู้ใช้งานระบุโดยวิธีการแบบสุ่ม [37]

ตารางที่ 3.2 บริการ EBI ที่ใช้ในการพัฒนาเวิร์คโฟลว์วิเคราะห์ของกุ้ง

ชื่อบริการ EBI	หน้าที่การทำงาน
transeq	ถอดรหัสจากลำดับนิวคลีโอไทด์ (ดีเอ็นเอ) เป็นลำดับกรดอะมิโน (โปรตีน)
garnier	ทำนายโครงสร้างขั้นที่สองของโปรตีน
emma_[DNA/Prot]	หาความคล้ายกันของลำดับ (Multiply alignment) ด้วยโปรแกรมประยุกต์ ClustalW
prettyplot	แสดงผลการเรียงของลำดับด้วยสีและเครื่องหมายบ่งความเหมือนและแตกต่าง
fseqboot_[DNA/Prot]	อัลกอริทึมในการทำ Bootstrapping replication
fdnapars	อัลกอริทึม DNA parsimony เพื่อเตรียมข้อมูลสำหรับการวิเคราะห์โครงสร้างต้นไม้สายวิวัฒนาการ
fprotpars	อัลกอริทึม Protein parsimony เพื่อเตรียมข้อมูลสำหรับการวิเคราะห์โครงสร้างต้นไม้สายวิวัฒนาการ
fconsense_[DNA/Prot]	หาโครงสร้างต้นไม้สายวิวัฒนาการที่มีความเป็นไปได้มากที่สุด (Majority-rule and strict consensus tree)

ตารางที่ 3.3 พารามิเตอร์ของแต่บริการหรือเว็บเซอร์วิสที่กำหนดในเวิร์คโฟลว์

พารามิเตอร์	คำอธิบาย	ค่าที่กำหนด
Database	ชื่อของฐานข้อมูลที่จะให้โปรแกรม BLAST เข้าไปค้นหาลำดับเบส	DDBJNEW
Frame3	หมายเลขเฟรม 3 สำหรับการถอดรหัส [13][14]	3
osformat_[DNA/Prot]	รูปแบบลำดับของเอาท์พุตสำหรับวิเคราะห์โครงสร้างต้นไม้สายวิวัฒนาการ	phylip
seqtype_[DNA/Prot]	รูปแบบของการทำ Bootstrapping replicate ของลำดับเบส	ดีเอ็นเอ = d โปรตีน= p
Reps_[DNA/Prot]	จำนวนครั้งของ Bootstrapping replicate ในสร้างข้อมูลซ้ำๆ	ผันแปรตามกรณีทดสอบ
rootTree_[DNA_Prot]	ให้สร้างโครงสร้างต้นไม้สายวิวัฒนาการแบบมีราก	Yes



รูปที่ 3.7 เวิร์คโฟลว์แรกที่พัฒนาขึ้นโดยใช้บริการต่างๆที่กระจายอยู่บนอินเตอร์เน็ต

อินพุตของเวิร์คโฟลว์คือลำดับนิวคลีโอไทด์ของกุ้งแชบ๊วยในรูปแบบ Fasta โดยได้จากห้องปฏิบัติการ ซึ่งมีจำนวน 17-25 ลำดับ แต่ลำดับประกอบด้วย 250 คู่เบสที่ได้มาจากส่วน Exon7-10 ของโครโมโซมกุ้ง ตัวอย่างอินพุตลำดับนิวคลีโอไทด์รูปแบบ Fasta ดังรูปที่ 3.8

```
>SRE26
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGGAGTTCAGGTACGGCAAGTAC
CTGGGCGAGGCCTTCCGCGGCAACAACCAGCTGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGAT
TGACCGGCATGACGCCCTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTGGAGGAGACA
TGATCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
>TDE8
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGGAGTTCAGGTACGGCAAGTAC
CTGGGCGAGGCCTTCCGCGGCAACAACCAGCTGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGAT
TGACCGGCATGACGCACTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTGGAGGAGACA
TGACCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
>SKE42
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGGAGTTCAGGTACGGCAAGTAC
CTGGGCGAGGCCTTCCGCGGCAACAACCAGCTGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGAT
TGACCGGCATGACGCACTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTGGAGGAGACA
TGATCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
```

รูปที่ 3.8 ตัวอย่างอินพุตลำดับนิวคลีโอไทด์รูปแบบ Fasta ของกุ้งแชบ๊วย

กระบวนการสร้างหรือทำนายโครงสร้างต้นไม้สายวิวัฒนาการ จะเรียกบริการ *fseqboot* สำหรับการทำ Bootstrapping replicate, *fdnapars* และ *fprotpars* สำหรับการทำงานของอัลกอริทึม DNA parsimony และ Protein parsimony ตามลำดับและ *fconsense* สำหรับการหาโครงสร้างต้นไม้สายวิวัฒนาการที่มีความเป็นไปได้มากที่สุด

ตารางที่ 3.4 แสดงผลการทดสอบของเวิร์คโฟลว์ด้วยจำนวนของการทำซ้ำ (Number of bootstrapping replicates) ที่จำนวนต่างๆคือ 100, 200, 500, 1,000 และ 2,000 ตามลำดับ และตรวจวัดเวลาในการทำงาน พบว่าสำหรับการกำหนด Bootstrapping replicates ที่จำนวน 2,000 ชุด ทำให้ Data flow ระหว่างบริการ fseqboot -> fdnapars และ fseqboot -> fprotpars ไม่สามารถทำงานได้เนื่องจากเกิดเกินเวลาหรือเวิร์คโฟลว์ไม่ตอบสนองในการทำงานใดๆ (Not responding) เกิดความไม่คงเส้นคงวาในการทำงานช่วงส่งถ่ายข้อมูลเอาท์พุตระหว่างกันของเว็บเซอร์วิสดังกล่าว

ข้อความแสดงความผิดพลาดดังกล่าวที่โปรแกรมทาเวอร์นารายงานในรูปที่ 3.9 หมายถึง มีปัญหาเกี่ยวกับการสื่อสารภายในบริการ Sopalab ของ EBI อาจเนื่องด้วยใช้เวลาในการค้นหาข้อมูลเกินค่าเวลาการทำงานโดยปริยาย (Default time-out) ภายในของบริการ EBI ที่ตั้งค่าไว้ ซึ่งเป็นปัญหาภายในของ EBI เอง เราไม่สามารถเข้าไปแก้ไขได้ จึงจะต้องหาแนวทางแก้ไขที่ฝั่งของเราเอง

ตารางที่ 3.4 กรณีทดสอบและผลการทดสอบ

จำนวนชุดข้อมูลของการทำซ้ำ (No. of bootstrapping replicates)	**เวลาที่ใช้ทำงาน (นาที)**
100	02.50
250	12.20
500	20.29
1,000	20.40
2,000	ล้มเหลว (เกินเวลา)

```
ServiceError Message="Task failed due to problem invoking soaplab service"
TimeStamp="Feb 6, 2008 11:27:19 AM"
org.embl.ebi.SoaplabShare.SoaplabException:
Internal communication failed. (in fetchResults)
```

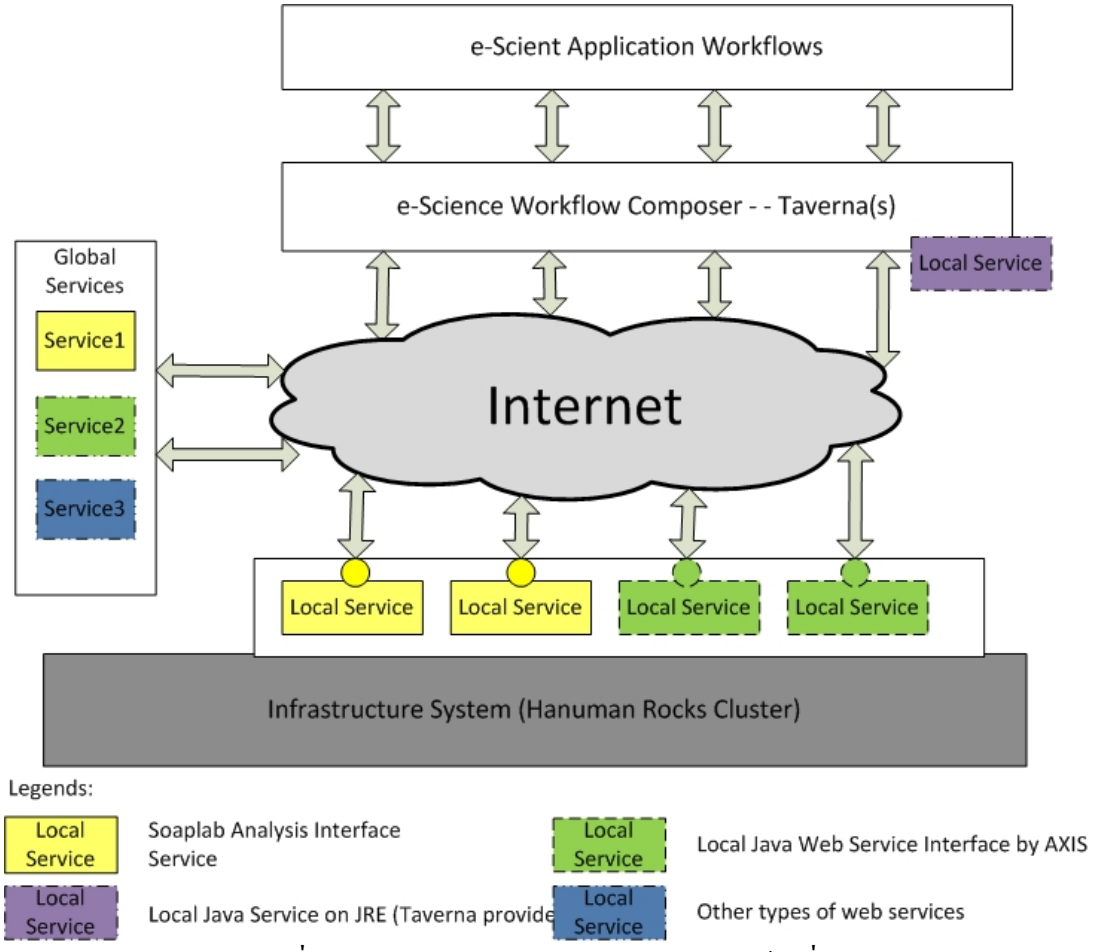
รูปที่ 3.9 ข้อความแสดงความผิดพลาดในการใช้บริการของ EBI ที่โปรแกรมทาเวอร์นารายงาน

3.4.4 การแก้ไขปัญหาการเดินเวลาและความไม่คงเส้นคงวาในการทำงาน

วิทยานิพนธ์นี้เสนอการสร้างบริการท้องถิ่น ในส่วนการทำนายหรือการสร้างโครงสร้างต้นไม้สายวิวัฒนาการเพื่อแก้ไขปัญหาการเกินเวลา และความไม่คงเส้นคงวาในการทำงาน บริการท้องถิ่นจะสามารถช่วยประหยัดเวลาในประมวลผล และให้ผลลัพธ์ที่ถูกต้องตรงตามความต้องมากที่สุด สถาปัตยกรรมโดยรวมของบริการท้องถิ่นที่ได้ออกแบบไว้แสดงได้ดังรูปที่ 3.10 โปรแกรมทาเวอร์นาสามารถทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล และสามารถเข้าถึงเว็บเซอร์วิสด้วยวิธีการแบบเวิร์คโฟลว์ได้หลากหลายประเภทผ่านระบบอินเตอร์เน็ต จากรูปที่ 3.10 แสดงบริการที่เกี่ยวข้องในการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ ที่สร้างเป็นบริการท้องถิ่น ได้แก่ บริการ Soaplab แทนด้วยบล็อกสีเหลือง และบริการ Java Web Service (JWS) แทนด้วยบล็อกสีเขียว บริการท้องถิ่นเหล่านี้จะทำงานบนระบบ Hanuman ซึ่งเป็นระบบ Rocks linux cluster

ของศูนย์กริดมหาวิทยาลัยสงขลานครินทร์ นอกจากนี้ ยังจะใช้ Java local service หรือบริการท้องถิ่นที่ใช้ความสามารถของ Java interpreter ของเครื่องคอมพิวเตอร์ส่วนบุคคลของผู้ใช้งานเอง ซึ่งแสดงด้วยบล็อกสีม่วง บริการอื่นๆที่ยังสามารถทำงานได้ถูกต้อง ไม่เกิดปัญหาการเกินเวลาและความไม่คงเส้นคงวาในการทำงาน ก็ยังคงเรียกใช้งานบริการเหล่านี้จากเครือข่ายอินเตอร์เน็ตเช่นเดิม (Global services)

เราจะใช้โปรแกรม Soaplab Analysis Tool เป็นเครื่องมือสร้างตัวเชื่อมประสานโปรแกรมประยุกต์ที่ทำนายโครงสร้างต้นไม้สายวิวัฒนาการให้เป็นเว็บเซอร์วิส เพื่อให้สามารถเชื่อมต่อเข้าไปในระบบเวิร์คโฟลว์ของโปรแกรมทาเวอร์นาได้ โดยสถาปัตยกรรมของโปรแกรมทาเวอร์นารองรับการเพิ่มเว็บเซอร์วิสอยู่แล้ว



รูปที่ 3.10 สถาปัตยกรรมโดยรวมของบริการท้องถิ่น

รูปที่ 3.11 แสดงกระบวนการสร้างโครงสร้างต้นไม้สายวิวัฒนาการเพิ่มเติมในวิทยานิพนธ์นี้โดยสร้างเว็บเซอร์วิสสองชนิด คือ โปรแกรมประยุกต์สำหรับการทำนายโครงสร้างต้นไม้สายวิวัฒนาการทั้งอัลกอริทึม Parsimony สำหรับลำดับนิวคลีโอไทด์และลำดับกรดอะมิโน โดยจะห่อหุ้มหรืออิมพลีเมนท์ด้วยโปรแกรม Soaplab Analysis Tool แทนด้วยบล็อกสีเหลือง ประกอบด้วยโปรแกรม fseqboot, fdnapars, fprotpars และ fconsense ส่วนอีกชนิดคือเว็บเซอร์วิสสำหรับการจัดการอินพุตและเอาท์พุตภายในเวิร์คโฟลว์ซึ่งจะพัฒนาเป็น Java Web Service (JWS) แทนด้วยบล็อกสีเขียว

JWS ประกอบด้วยบริการ *InputFileOperation* สำหรับทำหน้าที่ป้อนอินพุตข้อมูลลำดับนิวคลีโอไทด์เข้าสู่กระบวนการทำนายและบริการ *FecthResultOperation* สำหรับทำหน้าที่อ่านและส่งผลลัพธ์การทำงานของการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ กลับมาให้โปรแกรมทาเวอร์นาซึ่งจะได้ผลลัพธ์สองชนิดคือ โครงสร้างต้นไม้สายวิวัฒนาการที่อยู่ในรูปแบบ Treefile และอยู่ในรูปแบบ Text เช่น ตัวอย่างโครงสร้างต้นไม้สายวิวัฒนาการจากการทำนายจากการ Bootstrapping replicaton จำนวน 1,000 ชุดของลำดับนิวคลีโอไทด์ซึ่งอยู่ในรูปแบบ Treefile ดังรูปที่ 3.12 ซึ่งข้อมูลในรูปแบบ Treefile สามารถนำไปเปิดกับโปรแกรม Treeview ให้โปรแกรมสร้างเป็นโครงสร้างต้นไม้สายวิวัฒนาการ ตามประเภทที่ต้องการภายหลังได้หลายแบบได้แก่ Slanted Cladogram, Rectagular Cladogram และ Phylogram [44] ตัวอย่างโครงสร้างต้นไม้สายวิวัฒนาการจากการทำนายจากการทำ Bootstrapping replicate จำนวน 1,000 ชุดของลำดับนิวคลีโอไทด์ซึ่งอยู่ในรูปแบบ Text ดังรูปที่ 3.13

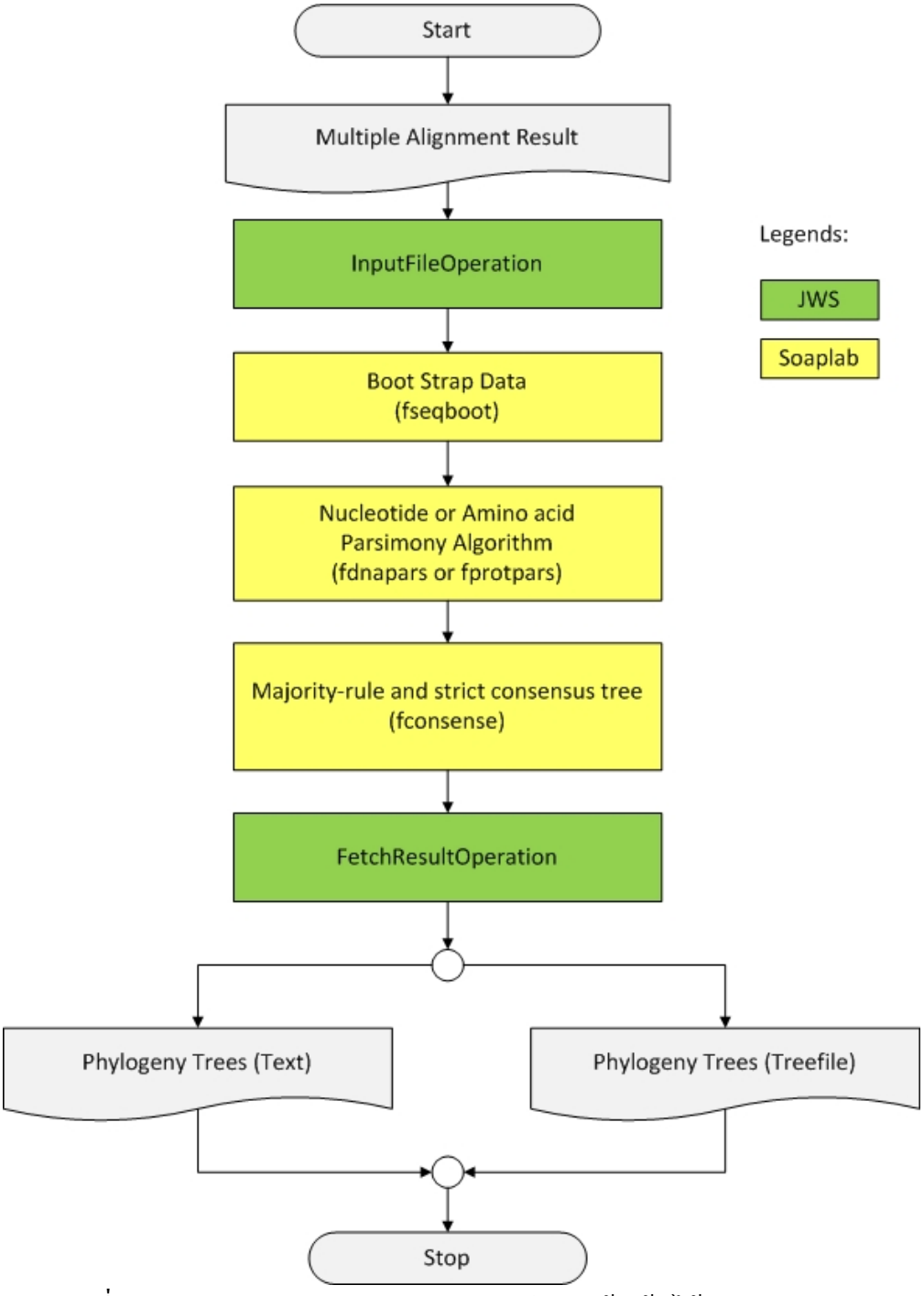
3.4.5 การพัฒนาบริการท้องถิ่น

ขั้นตอนการพัฒนาบริการท้องถิ่น สำหรับการทำนายโครงสร้างสายวิวัฒนาการของกุ้งเพื่อแก้ปัญหาการเกินเวลาและความไม่คงเส้นคงวาในการทำงานงานมีดังนี้

1) การใช้งานโปรแกรม PHYLIB

โปรแกรม PHYLIB เป็นซอฟต์แวร์ชนิด Open source ของชุดโปรแกรม Emboss ซึ่งเป็นชุดเดียวกันกับที่ทำงานเป็นเว็บเซอร์วิสที่ให้บริการโดย EBI วิทยานิพนธ์นี้ติดตั้งโปรแกรม

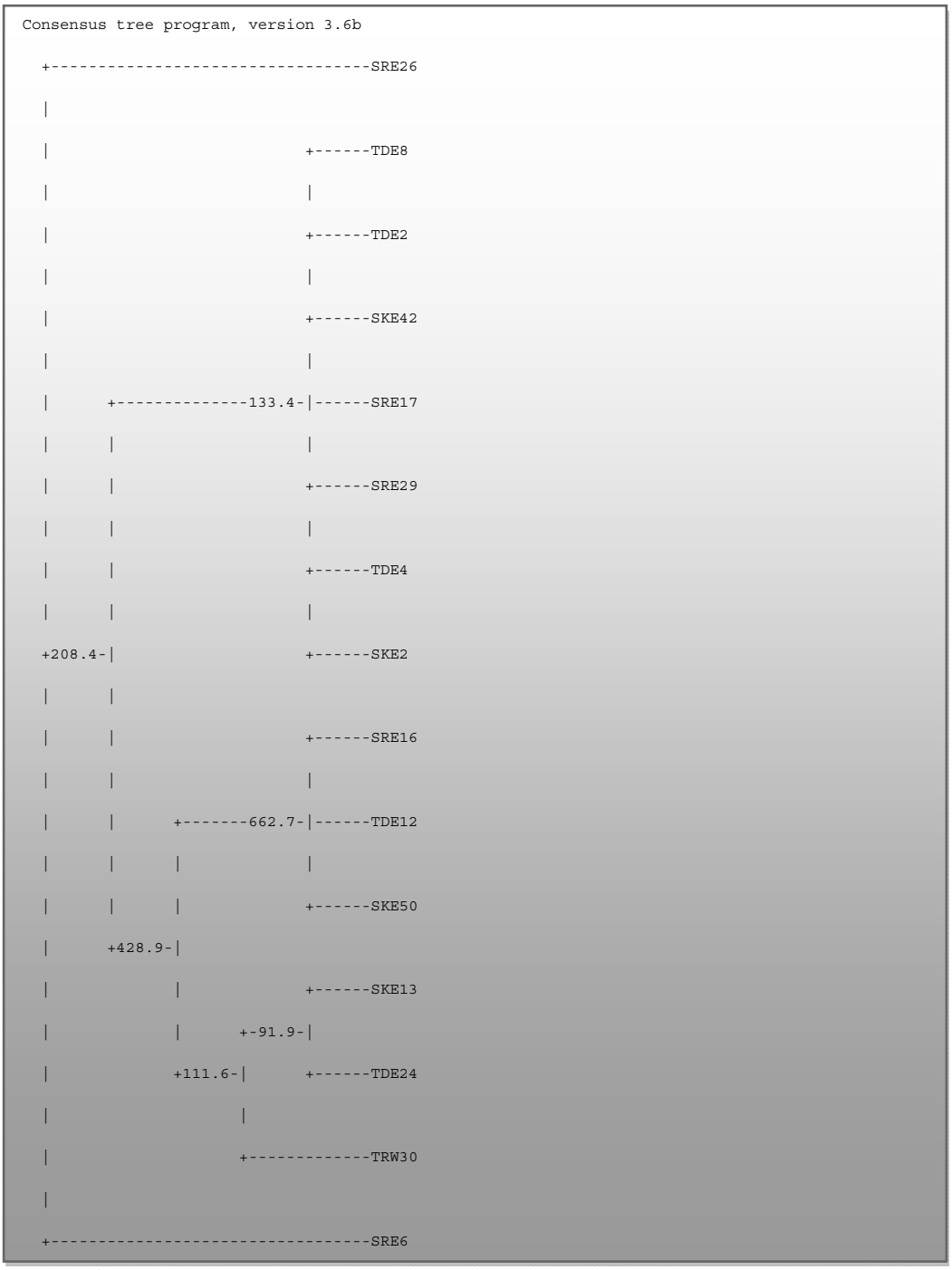
PHYLIB รุ่น 3.6b [48] สำหรับการทำนายโครงสร้างต้นไม้สายวิวัฒนาการของสิ่งมีชีวิตบนระบบคอมพิวเตอร์คลัสเตอร์ Hanuman ซึ่งใช้ระบบปฏิบัติการ Rocks Linux Cluster รุ่น 4.3 และใช้ Apache Tomcat รุ่น 5.0.28 เป็น Web service container



รูปที่ 3.11 Flowchart ของกระบวนการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ

```
(SRE26:1000.0,((TDE8:1000.0,TDE2:1000.0,SKE42:1000.0,SRE17:1000.0,SRE29:1000.0,
TDE4:1000.0,SKE2:1000.0):133.4,((SRE16:1000.0,TDE12:1000.0,SKE50:1000.0):662.7,
((SKE13:1000.0, TDE24:1000.0):91.9,TRW30:1000.0):111.6):428.9):208.4,SRE6:1000.0);
```

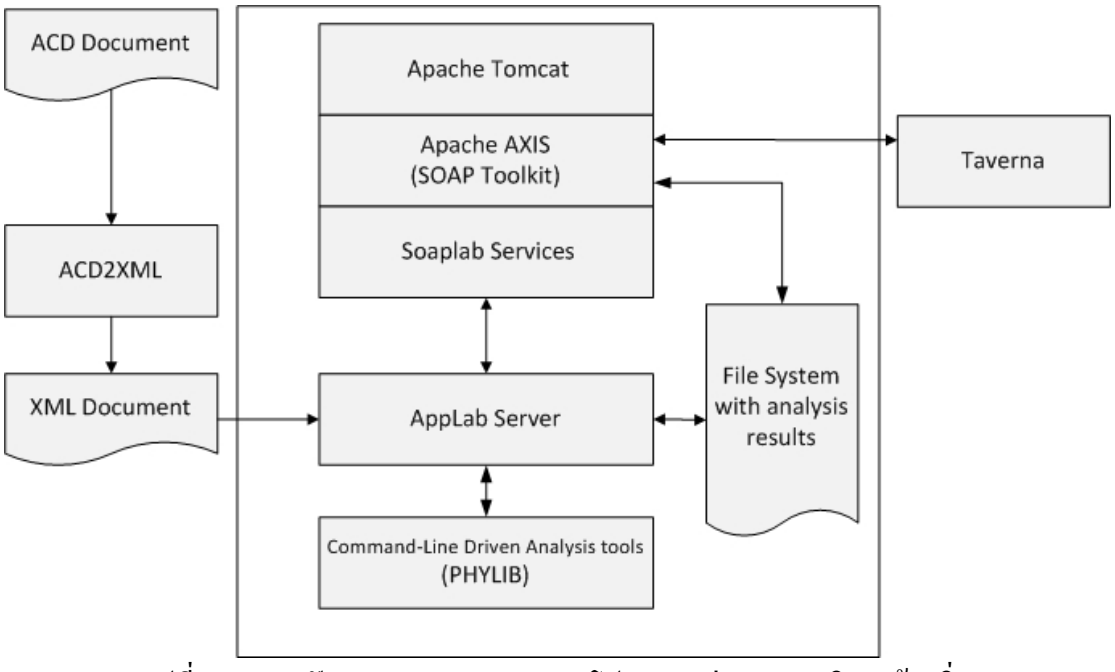
รูปที่ 3.12 โครงสร้างต้นไม้สายวิวัฒนาการในรูปแบบ Treefile ของลำดับนิวคลีโอไทด์



รูปที่ 3.13 โครงสร้างต้นไม้สายวิวัฒนาการในรูปแบบ Text ของลำดับนิวคลีโอไทด์

2) ติดตั้งโปรแกรม Soaplab Analysis Tool และ SOAP Toolkit

สถาปัตยกรรมของโปรแกรม Soaplab Analysis Tool ประกอบด้วยซอฟต์แวร์หลายชนิดประกอบกันเพื่อให้สามารถทำงานได้ตามต้องการดังที่ได้อธิบายแล้วในบทที่ 2 รูปที่ 3.14 แสดงโปรแกรม PHYLIB ในสถาปัตยกรรมการทำงานร่วมกับโปรแกรม Soaplab Analysis Tool แสดงให้เห็นถึงตำแหน่งการทำงานของโปรแกรม PHYLIB ในระบบการทำงานแบบเว็บเซอร์วิส ซึ่งจะติดต่อกับ Tool ของ Soaplab คือ AppLab Server โดย AppLab Server จะให้พอยต์เตอร์ของโปรแกรมต่างๆกับ Tool ของ Soaplab อีกโปรแกรมคือ Soaplab Services และ Tool ทั้งสองโปรแกรมทำงานบนพื้นฐานของ Apache Tomcat และ Apache AXIS ซึ่งเป็น Soap Toolkit ทำหน้าที่เชื่อมประสานการติดต่อสื่อสารออกไปสู่โลกภายนอก ในวิทยานิพนธ์นี้ก็คือระบบเวิร์คโฟลว์ในโปรแกรมทาเวอร์นา นอกจากนี้ Apache AXIS ยังทำหน้าที่เป็นโฮสท์ให้กับบริการ JWS ด้วย โปรแกรม Soaplab Analysis Tool มี Binary package ที่รวบรวม Tool เหล่านี้ไว้เรียบร้อยแล้วคือ Analysis Interface [37]



รูปที่ 3.14 สถาปัตยกรรมการทำงานของโปรแกรมต่างๆของบริการท้องถิ่น

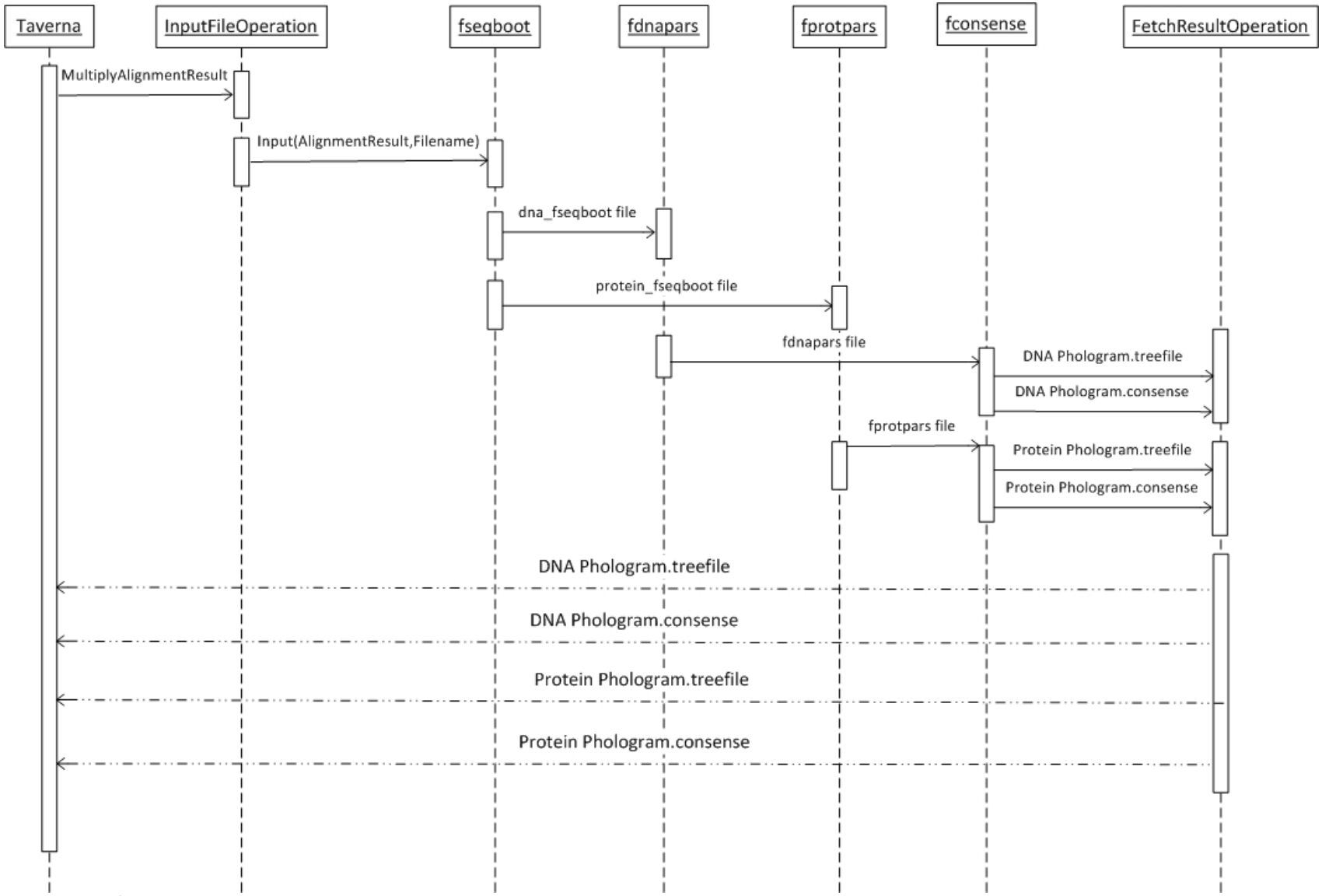
3) การสร้างตัวเชื่อมประสานเว็บเซอร์วิส (Web service interface)

การสร้างตัวเชื่อมประสานเว็บเซอร์วิสหรือ Web service interface ให้กับชุดโปรแกรมในการทำนายโครงสร้างต้นไม้สายวิวิฒนาการ สามารถสร้างได้ตามสถาปัตยกรรมการทำงานในรูปที่ 3.14 เครื่องมือของ Soaplab Analysis Tool ที่สำคัญอีกตัวหนึ่งคือ ACD2XML ซึ่งทำหน้าที่ในการแปล (translate) เอกสาร AJAX Command Definition (ACD) [49] เป็นเอกสาร XML เนื่องจากเอกสาร XML เป็นภาษามาตรฐานของโปรโตคอล SOAP ซึ่งเป็นหัวใจของการทำงาน ดังนั้นขั้นตอนสำคัญคือ การสร้างเอกสาร ACD เพื่ออธิบายการทำงานและการปรับแต่งของชุดโปรแกรม PHYLIB ที่อยู่ในรูปแบบ Interactive command line ให้เข้ากับการใช้งานในงานวิทยานิพนธ์คือ Non-interactive command line จากนั้นจึงใช้ ACD2XML แปลเอกสาร ACD เป็นเอกสาร XML เพื่อส่งต่อเข้าสู่กระบวนการทำงานของ Soaplab Analysis Tool ต่อไป

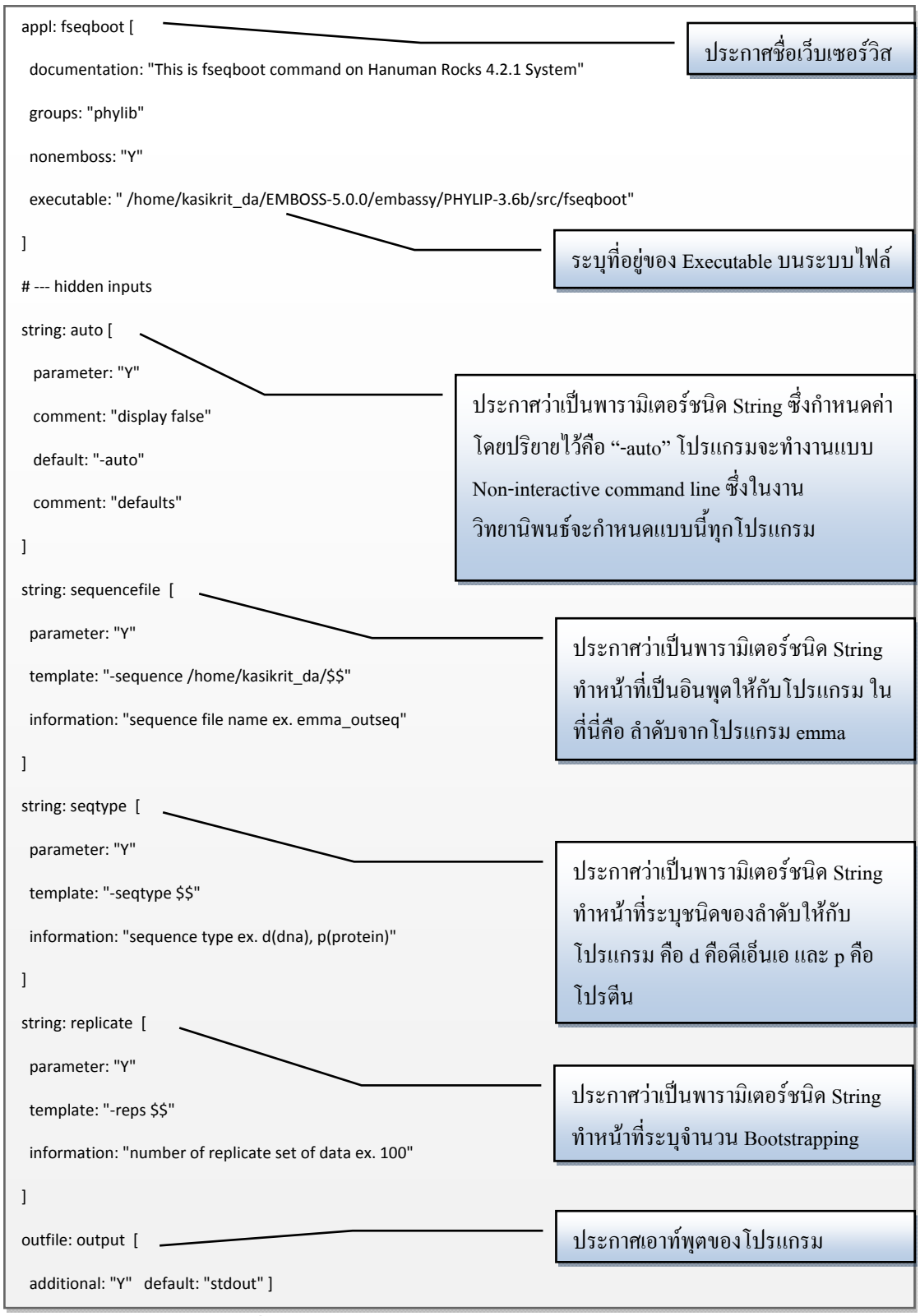
ชุดโปรแกรม PHYLIB สำหรับการทำงานโครงสร้างสายต้นไม้วิวัฒนาการประกอบด้วยโปรแกรม fseqboot, fdnapars, fprotpars และ fconsense ซึ่งลำดับการทำงานภายในเวิร์คโฟลว์ของโปรแกรมทาเวอร์นาและการส่งต่อข้อมูลสามารถอธิบายด้วย Sequence Diagram ดังรูปที่ 3.15 โดยประกอบด้วยการปรับแต่งการทำงานด้วยเอกสาร ACD ดังต่อไปนี้

- **เอกสาร ACD สำหรับโปรแกรม fseqboot**

โปรแกรม fseqboot สามารถรับอินพุตได้ทั้งลำดับนิวคลีโอไทด์และลำดับกรดอะมิโน ซึ่งการแยกประเภทของลำดับกำหนดไว้เป็นพารามิเตอร์ ยิ่งกำหนดให้โปรแกรม fseqboot ทำ Bootstrapping replicate มากเท่าไร ก็จะได้โครงสร้างต้นไม้สายวิวัฒนาการที่มีความเป็นไปได้ และถูกต้องมากยิ่งขึ้น ผลลัพธ์ที่ได้จะส่งต่อให้โปรแกรม fdnapars สำหรับวิเคราะห์ลำดับนิวคลีโอไทด์ และ fprotpars สำหรับวิเคราะห์ลำดับกรดอะมิโน โดยในวิทยานิพนธ์นี้จะทดสอบจำนวน Bootstrapping replicate สูงสุดถึง 50,000 ชุดเพื่อการทดสอบประสิทธิภาพของบริการท้องถิ่น เอกสาร ACD สำหรับโปรแกรม fseqboot ดังรูปที่ 3.16



รูปที่ 3.15 Sequence diagram ของการทำงานภายในเวิร์คโฟลว์ของโปรแกรมทาเวอร์นาในการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ



รูปที่ 3.16 เอกสาร ACD สำหรับโปรแกรม fseqboot

- **เอกสาร ACD สำหรับโปรแกรม fdnapars**

โปรแกรม fdnapars มีอัลกอริทึมการทำงาน DNA Parsimony สำหรับการเตรียมข้อมูลเพื่อส่งต่อให้กับโปรแกรม fconsense ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์ เอกสาร ACD สำหรับโปรแกรม fdnpars ดังรูปที่ 3.17

```
appl: fdnapars [
  documentation: "This is fdnapars command on Hanuman Rocks 4.2.1 System"
  groups: "phylib"
  nonemboss: "Y"
  executable: "/home/kasikrit_da/EMBOSS-5.0.0/embassy/PHYLIP-3.6b/src/fdnapars"
]
# --- hidden inputs
string: auto [
   parameter: "Y"
   comment: "display false"
   default: "-auto"
   comment: "defaults"
]
string: fseqbootfile  [
  parameter: "Y"
  template: "-sequence /home/kasikrit_da/analysis-interfaces/a/unknown/Projects/default/Data/$$.fseqboot"
  information: "sequence file name ex. emma_outseq"
]
outfile: output  [
  additional: "Y"
  default: "stdout"
]
```

ระบุที่อยู่ของ Executable บนระบบไฟล์

ระบุที่อยู่ของอินพุตให้กับโปรแกรม fdnpars ซึ่งก็คือเอาท์พุตของโปรแกรม fseqboot ของลำดับนิวคลีโอไทด์

รูปที่ 3.17 เอกสาร ACD สำหรับโปรแกรม fdnpars

- **เอกสาร ACD สำหรับโปรแกรม fprotpars**

โปรแกรม fprotpars มีอัลกอริทึมการทำงาน Protein parsimony สำหรับการเตรียมข้อมูลเพื่อส่งต่อให้กับโปรแกรม fconsense ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับกรดอะมิโนต่อไป เอกสาร ACD สำหรับโปรแกรม fprotpars ดังรูปที่ 3.18

```
appl: fprotpars [
  documentation: "This is fprotpars command on Hanuman Rocks 4.2.1 System"
  groups: "phylib"
  nonemboss: "Y"
  executable: "/home/kasikrit_da/EMBOSS-5.0.0/embassy/PHYLIP-3.6b/src/fprotpars"
]
# --- hidden inputs
string: auto [
   parameter: "Y"
   comment: "display false"
   default: "-auto"
   comment: "defaults"
]
string: fseqbootfile  [
  parameter: "Y"
  template: "-sequence /home/kasikrit_da/analysis-interfaces/a/unknown/Projects/default/Data/$$.fseqboot"
  information: "sequence file name ex. emma_outseq"
]
outfile: output  [
  additional: "Y"
  default: "stdout"
]
```

ระบุที่อยู่ของ Executable บนระบบไฟล์

ระบุที่อยู่ของอินพุตให้กับโปรแกรม fdnpars ซึ่งก็คือเอาท์พุตของโปรแกรม fseqboot ของลำดับกรดอะมิโน

รูปที่ 3.18 เอกสาร ACD สำหรับโปรแกรม fprotpars

- **เอกสาร ACD สำหรับโปรแกรม fconsense**

โปรแกรม fonsense สามารถรับอินพุตได้ทั้งลำดับนิวคลีโอไทด์และลำดับกรดอะมิโนซึ่งสามารถแยกแยะได้โดยการระบุพารามิเตอร์ โปรแกรม fconsense ใช้ทำนายโครงสร้างต้นไม้สายวิวัฒนาการที่มีความเป็นไปได้มากที่สุด เอกสาร ACD สำหรับโปรแกรม fconsense ดังรูปที่ 3.19 และให้เอาท์พุตเป็นข้อมูลโครงสร้างต้นไม้สายวิวัฒนาการ 2 รูปแบบคือ Treefile และ Text ดังที่ได้กล่าวแล้ว

```
appl: fconsense [
  documentation: "This is fconsense command on Hanuman Rocks 4.2.1 System"
  groups: "phylib"
  nonemboss: "Y"
  executable: "/home/kasikrit_da/EMBOSS-5.0.0/embassy/PHYLIP-3.6b/src/fconsense"
]
# --- hidden inputs
string: auto [
   parameter: "Y"
   comment: "display false"
   default: "-auto"
   comment: "defaults"
]
string: root [
   parameter: "Y"
   comment: "display false"
   default: "-root Yes"
   comment: "defaults"
]
string: intreefile  [
  parameter: "Y"
  template: "-intreefile /home/kasikrit_da/analysis-interfaces/a/unknown/Projects/default/Data/$$.treefile"
  information: "sequence file name ex. emma_outseq"
]
outfile: output  [
  additional: "Y"
  default: "stdout"
]
```

ระบุที่อยู่ของ Executable บนระบบไฟล์

ระบุพารามิเตอร์ว่า ให้โปรแกรมทำนายโครงสร้างต้นไม้สายวิวัฒนาการชนิดมีรากร่วมกัน

ระบุที่อยู่ของอินพุตให้กับโปรแกรม fconsense ซึ่งก็คือเอาท์พุตของโปรแกรม fdnapars และ fprotpars นั่นเอง

รูปที่ 3.19 เอกสาร ACD สำหรับโปรแกรม fconsense

4) สร้างบริการท้องถิ่นสำหรับการจัดการด้านอินพุตและเอาท์พุต

บริการท้องถิ่นที่สร้างในงานวิทยานิพนธ์เป็นชนิด JWS ซึ่งเป็น Java class ทำหน้าที่จัดการด้านอินพุตและเอาท์พุตของการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ บนระบบไฟล์ตามที่ได้กำหนดไว้ในเอกสาร ACD ส่วนของ Java Class ที่เป็นแกนสำคัญของการทำงานเป็นดังนี้

- **บริการท้องถิ่นสำหรับการจัดการอินพุต**

บริการท้องถิ่นสำหรับการจัดการอินพุต ทำหน้าที่ป้อนข้อมูลอินพุตซึ่งเป็นลำดับนิวคลีโอไทด์และโปรตีนที่เป็นผลลัพธ์จากโปรแกรม Emma ของสถาบัน EBI ซึ่งทำงานอยู่บนเครือข่ายอินเตอร์เน็ตเข้าสู่กระบวนการทำนายโครงสร้างต้นไม้สายวิวัฒนาการ ซึ่งทำงานในระบบเวิร์คโฟลว์ของโปรแกรมทาเวอร์นาดังรูปที่ 3.20

```
//InputFile Operation
        public String InputFileOperation(String contents, String filename) throws IOException{
                try {
                        FileWriter fw = new FileWriter(filename);
                        PrintWriter pout = new PrintWriter(new BufferedWriter(fw));
                        pout.print(contents);
                        pout.close();
    }
                catch (IOException ioe) { return "IO error:" + ioe ;}
                return filename;
        }
```

พารามิเตอร์ชื่อไฟล์ระบุโดยผู้ใช้งาน

พารามิเตอร์ที่บอกว่าเป็นเอาท์พุตจากโปรแกรม Emma หรือการทำ Multiply alignment นั่นเอง

รีเทิร์นชื่อไฟล์เพื่อเป็นพอยต์เตอร์กับโปรเซสอื่นๆ

รูปที่ 3.20 Java class ของบริการท้องถิ่นสำหรับการจัดการอินพุต

- **บริการท้องถิ่นสำหรับการจัดการเอาท์พุต**

บริการท้องถิ่นสำหรับการจัดการเอาท์พุต ทำหน้าที่อ่านผลลัพธ์ของการทำนายโครงสร้างสายวิวัฒนาการทั้งของลำดับนิวคลีโอไทด์และโปรตีน จากระบบไฟล์แล้วส่งผลลัพธ์กลับมาให้ที่ระบบเวิร์คโฟลว์ในโปรแกรมทาเวอร์นาดังรูปที่ 3.21 และ รูปที่ 3.22

1) Java class สำหรับการอ่านและส่งผลลัพธ์ของรูปแบบ Treefile

```
//Reading a file and show content of emboss result in analysis_interface dir

public String ReadFileEmbossTreefile(String filename)throws IOException{

                String s, content="";

                try {

FileReader fin = new FileReader("/home/kasikrit_da/analysis-
interfaces/a/unknown/Projects/default/Data/"+filename+".treefile");

                BufferedReader bin = new BufferedReader(fin);

                while( (s = bin.readLine()) != null){

                        content += s + "\n";;

                }

                bin.close();

                }

                catch (IOException ioe) {

        System.out.println( "IO error:" + ioe );

            }

                return content;

}
```

ที่อยู่ของไฟล์ผลลัพธ์รูปแบบ Treefile บนระบบไฟล์

ส่งผลลัพธ์มาให้โปรแกรมทาเวอร์นา

รูปที่ 3.21 Java class สำหรับการอ่านและส่งผลลัพธ์ของรูปแบบ Treefile

2) Java class สำหรับการอ่านและส่งผลลัพธ์ในรูปแบบ Text

```
public String ReadFileEmbossConsense(String filename)throws IOException{
        String s, content="";
        try {
FileReader fin = new FileReader("/home/kasikrit_da/analysis-
interfaces/a/unknown/Projects/default/Data/"+filename+".fconsense");
        BufferedReader bin = new BufferedReader(fin);
        while( (s = bin.readLine()) != null){
                content += s + "\n";
        }
        bin.close();
        }
        catch (IOException ioe) {
    System.out.println( "IO error:" + ioe );
   }
        return content;
}
```

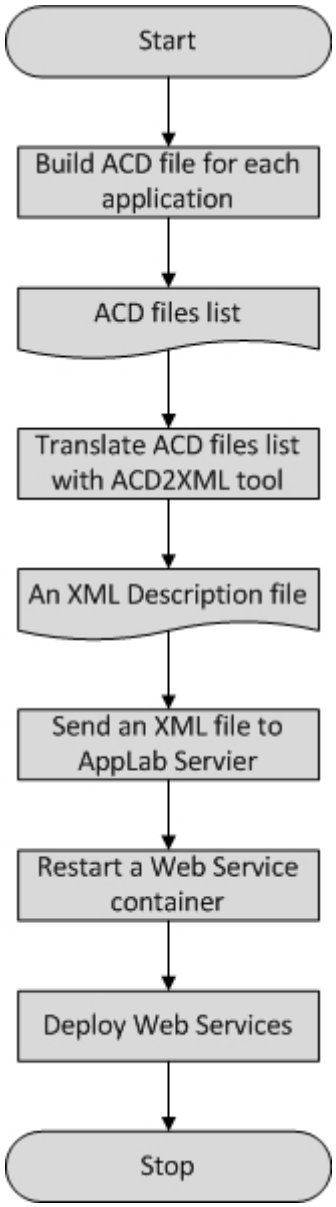
ที่อยู่ของไฟล์ผลลัพธ์รูปแบบ Text บนระบบไฟล์

ส่งผลลัพธ์มาให้โปรแกรมทาเวอร์นา

รูปที่ 3.22 Java class สำหรับการอ่านและส่งผลลัพธ์ในรูปแบบ Text

5) การใช้งานบริการท้องถิ่น

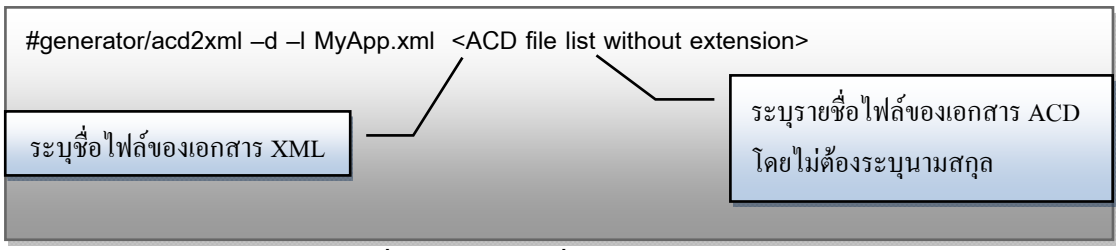
การใช้งานบริการท้องถิ่น (Local web services employing) ด้วยโปรแกรม Soaplab Analysis Tool โดยมี Flowchart แสดงขั้นตอนดังรูปที่ 3.23 และอธิบายการใช้งานบริการท้องถิ่นดังนี้



รูปที่ 3.23 ขั้นตอนการใช้งานบริการท้องถิ่นด้วยโปรแกรม Soaplab Analysis Tool

5.1) เอกสาร ACD จะต้องประกอบด้วยนามสกุล acd ได้แก่ fseqboot.acd, fdnapars.acd, fprotpars.acd และ fconsense.acd จากนั้นบันทึกเอกสาร ACD ที่ระบบไฟล์ soaplab/src/etc/acd/applab หรือ analysis-interfaces/metadata สำหรับชุดโปรแกรมแบบ Binary package

5.2) แปลงเอกสาร ACD เป็นเอกสาร XML ด้วยเครื่องมือ ACD2XML ดังรูปที่ 3.24



รูปที่ 3.24 การใช้เครื่องมือ ACD2XML

5.3) ส่งเอกสาร XML ให้กับ AppLab Server โดยให้ทำงานเป็นโปรเซสเบื้องหลัง

#run-AppLab-server MyApp.xml &

5.5) รีสตาร์ทโปรเซสของ Apache Tomcat

5.6) เรียกใช้งานเซอร์วิสโดยให้ทำงานเป็นโปรเซสเบื้องหลัง

#./ws/deploy-web-services –d –A MyApp.xml &

5.7) เอ็นพอยต์ของเว็บเซอร์วิสคือ http://<hostname>:<port>/axis/services ในวิทยานิพนธ์คือ http://hanuman.psu.ac.th:8081/axis/services ซึ่งสามารถตรวจสอบการเข้าถึงกับเว็บเบราเซอร์และโปรแกรมทาเวอร์นา

5.8) สำหรับบริการท้องถิ่นชนิด JWS สามารถเรียกใช้งานเซอร์วิสได้โดยบันทึก Java Class เป็นนามสกุล .jws แล้วไว้ที่ระบบไฟล์ <tomcat dir>/webapps/axis ในวิทยานิพนธ์นี้คือ tomcat-5.0.28/webapps/axis และเอ็นพอยต์ในการเข้าถึงบริการคือ http://<hostname>:<port>/axis/<java class name>.jws?wsdl ในที่นี้คือ http://hanuman.psu.ac.th:8081/axis /HanumanWSDL.jws ?wsdl

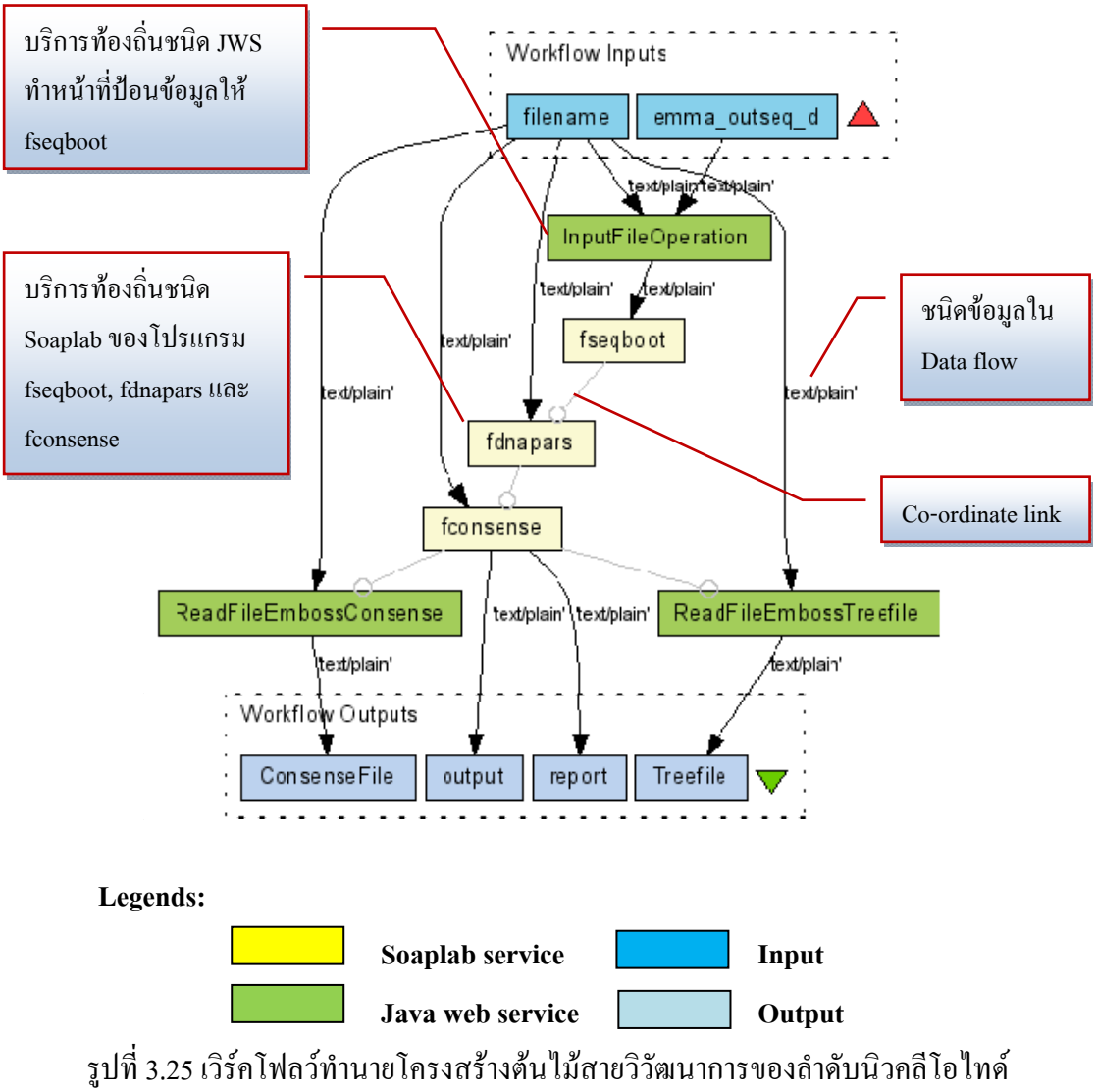
5.9) การรีสตาร์ทโปรเซสของ Apache Tomcat หลังจากกระบวนการการเรียกใช้งานเซอร์วิสสามารถทำได้โดยอิสระ ไม่กระทบการทำงานของบริการต่อที่ทำงานเป็นโปรเซสเบื้องหลัง แต่การเรียกใช้งานเซอร์วิสใหม่ จำเป็นต้องรีสตาร์ทโปรเซสของ Apache Tomcat ทุกครั้งเพื่อให้ระบบรับรู้ถึงบริการใหม่ที่เพิ่มเข้ามา

3.4.6 การพัฒนาเวิร์คโฟลว์ที่ใช้บริการท้องถิ่น

บริการท้องถิ่นที่สร้างขึ้นทั้งบริการ Soaplab และบริการ JWS สามารถเข้าถึงได้ด้วยโปรแกรมทาเวอร์นาและสามารถสร้าง Instance ของบริการเข้าไปในระบบเวิร์คโฟลว์เพื่อการทำนายโครงสร้างต้นไม้สายวิวัฒนาการได้ ดังเวิร์คโฟลว์ที่จะกล่าวถึงต่อไปนี้

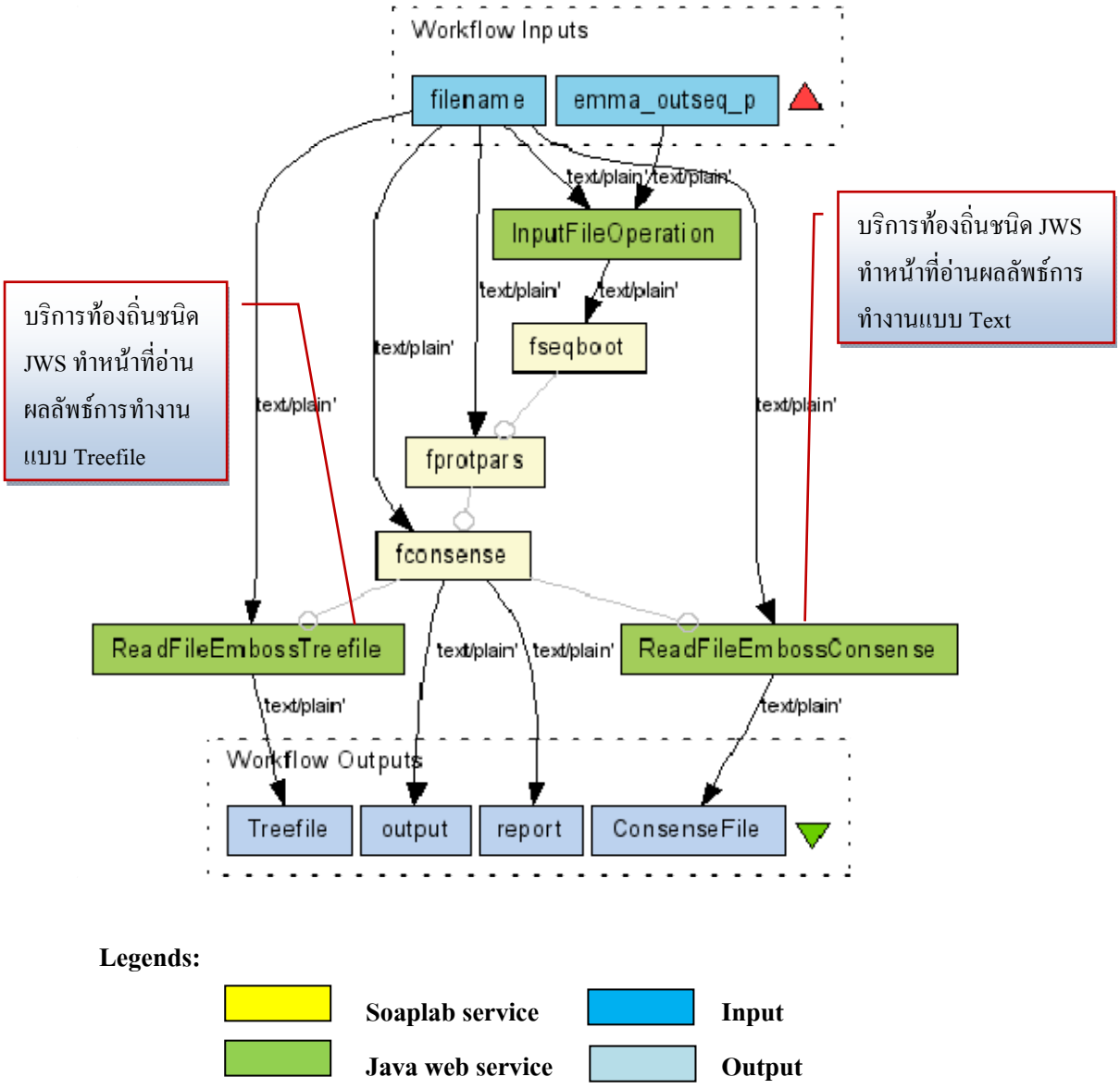
- **การพัฒนาเวิร์คโฟลว์ของแต่ละกระบวนการทำนาย**

รูปที่ 3.25 แสดงเวิร์คโฟลว์ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์เพราะมีการใช้งานบริการท้องถิ่น fdnapars ในการทำงานอัลกอริทึม DNA parsimony สังเกตได้ว่ามีการใช้ Co-ordinate link หรือการเชื่อมโยงแบบควบคุม คือบริการถัดไปจะเริ่มทำงานได้ก็ต่อเมื่อบริการก่อนหน้าทำงานเสร็จสิ้นเสียก่อน



รูปที่ 3.25 เวิร์คโฟลว์ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์

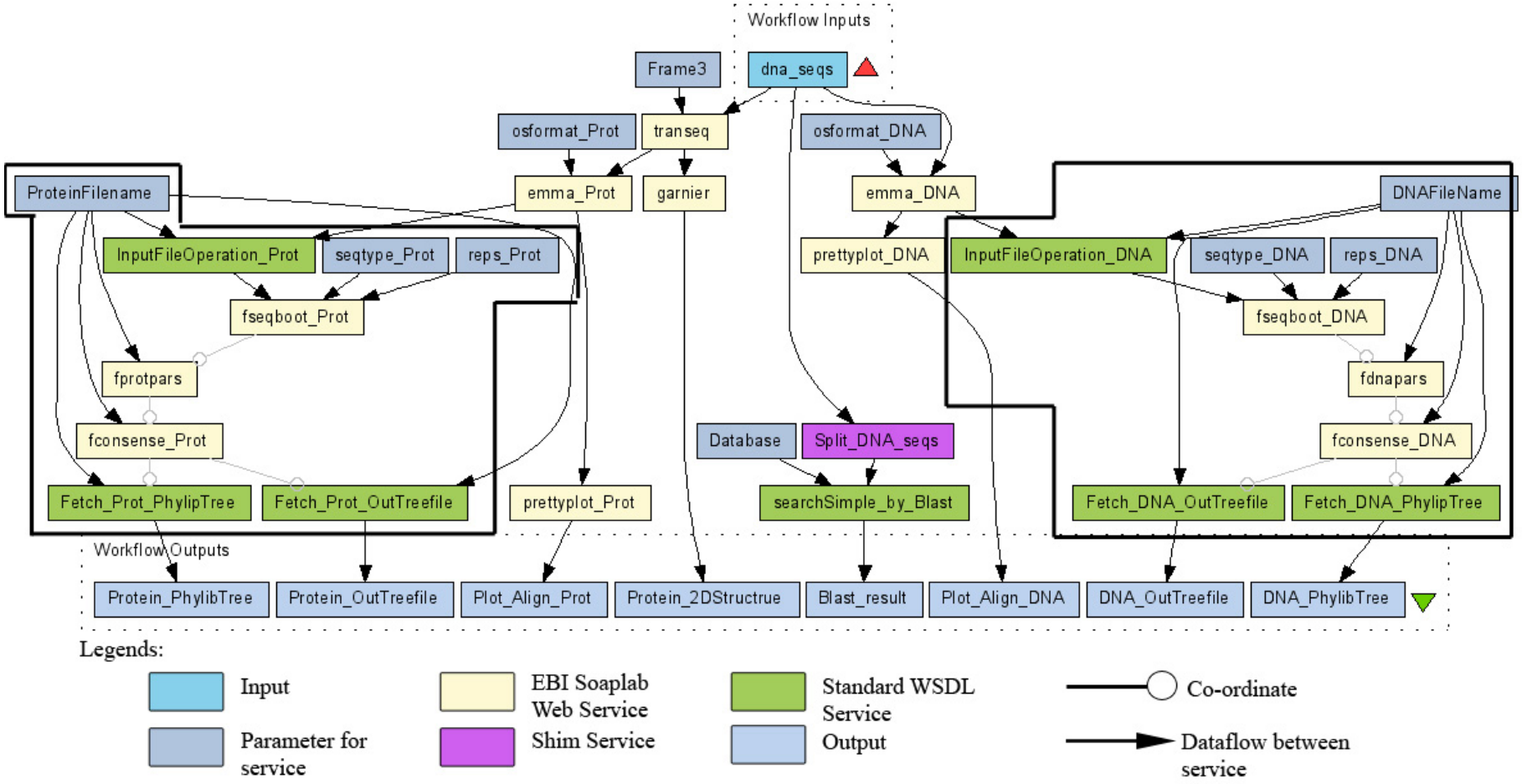
รูปที่ 3.26 เวิร์คโฟลว์ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับกรดอะมิโนเพราะมีการใช้งานบริการท้องถิ่น fprotpars ในการทำงานอัลกอริทึม Protein parsimony จะสังเกตว่ามีการใช้ Co-ordinate link หรือการเชื่อมโยงแบบควบคุมเช่นเดียวกัน



รูปที่ 3.26 เวิร์คโฟลว์ทำนายโครงสร้างต้นไม้สายวิวัฒนาการของลำดับกรดอะมิโน

- **การพัฒนาเวิร์คโฟลว์โดยรวบรวมระบบงานทั้งหมดเข้าด้วยกัน**

จากการทดสอบการทำงานอย่างง่ายก็ทำให้ทราบว่าเวิร์คโฟลว์ที่สร้างในรูปที่ 3.25 และ รูปที่ 3.26 มีลำดับกระบวนการทำงาน การเข้าจังหวะการทำงานและมี Data flow ไหลในเวิร์คโฟลว์ได้ถูกต้องสมบูรณ์ดีแล้ว จึงทำการปรับแก้เวิร์คโฟลว์หลักเวิร์คโฟลว์แรก (รูปที่ 3.7) โดยเปลี่ยนมาใช้บริการท้องถิ่นที่ได้ทำการสร้างขึ้นมาใช้งานเอง จึงได้เวิร์คโฟลว์สำหรับการทำนายโครงสร้างต้นไม้สายวิวัฒนาการของกุ้งที่ใช้บริการท้องถิ่น (ในกรอบสีดำ) ดังรูปที่ 3.27



รูปที่ 3.27 เวิร์คโฟลว์ใหม่ที่ใช้บริการท้องถิ่น (ในกรอบสีดำ)

3.4.7 ผลการทดลอง

ตารางที่ 3.5 แสดงผลการทดสอบของกรณีทดสอบต่างๆของเวิร์คโฟลว์ใหม่ที่ใช้บริการท้องถิ่นในรูปที่ 3.27 โดยใช้จำนวน Bootstrapping replicate ของลำดับนิวคลีโอไทด์และกรดอะมิโนโดยจะเพิ่มขึ้นเรื่อยๆจนกระทั้งถึง 50,000 ชุด เวลาที่เวิร์คโฟลว์ทำงานแสดงในหน่วยนาทีซึ่งได้จากรายงานสถานะการทำงานจากโปรแกรมทาเวอร์นา

ตารางที่ 3.5 กรณีทดสอบและผลการทดสอบของเวิร์คโฟลว์ใหม่ที่ใช้บริการท้องถิ่น

จำนวนชุดข้อมูลของการทำซ้ำ (No. of Replicates)	**เวลาที่ใช้ในการทำงาน (นาที) (Time Used)**
100	02.12
250	02.06
500	02.06
1,000	02.06
2,000	02.24
3,000	03.04
4,000	03.47
5,000	03.52
10,000	08.09
20,000	15.08
30,000	22.02
40,000	29.22
50,000	36.17

ตารางที่ 3.6 แสดงการเปรียบเทียบกรณีทดสอบและผลการทดสอบของวิธีการทำงานแบบต่างๆของลำดับนิวคลีโอไทด์คือ วิธีการแบบคัดลอกแล้ววาง (Manual) กับวิธีอัตโนมัติ

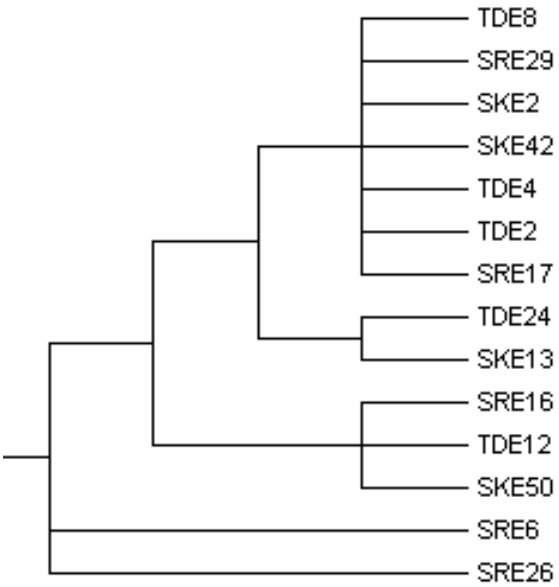
ด้วยการใช้เวิร์คโฟลว์ โดยมีสองประเภทคือ วิธีการแบบเวิร์คโฟลว์ที่ไม่ได้ใช้บริการท้องถิ่น และวิธีการแบบเวิร์คโฟลว์โดยใช้บริการท้องถิ่น สำหรับการทำ Bootstrapping replicate 1,000 ชุด เวิร์คโฟลว์แรกใช้เวลาทำงาน 2.06 นาที แต่ล้มเหลวในการทำงานเนื่องจากสาเหตุการเกินเวลาดังที่กล่าวแล้วที่จำนวนชุดข้อมูลของการทำ Bootstrapping replicate 2,000 ชุด วิธีการแบบคัดลอกและวางโดยนักชีวสารสนเทศใช้เวลาประมาณ 15 นาที ในขณะที่วิธีการทำงานแบบอัตโนมัติด้วยเวิร์คโฟลว์ที่ใช้บริการท้องถิ่นใช้เวลา 2.24 นาที ดังนั้นการลดลงของเวลาในการประมวลผลไปเป็นอย่างมีนัยสำคัญ เมื่อเราเพิ่มตัวเลขการทำ Bootstrapping replicate ของลำดับนิวคลีโอไทด์เป็น 20,000 และ 50,000 ชุด เวิร์คโฟลว์ที่ใช้บริการท้องถิ่นใช้เวลาทำงาน 15.08 และ 36.17 นาทีตามลำดับ วิทยานิพนธ์นี้ไม่ได้ทดสอบกรณีทดสอบนี้โดยวิธีการคัดลอกแล้ววางเนื่องจากทดสอบทำได้ยากมากหรือเป็นไปไม่ได้ที่จะคัดลอกแล้ววางข้อมูลจำนวนมากเหล่านี้ ระหว่างเว็บเพจของกระบวนการต่างๆ

ตารางที่ 3.6 ตารางการเปรียบเทียบกรณีทดสอบและผลการทดสอบของวิธีการทำงานแบบต่างๆ

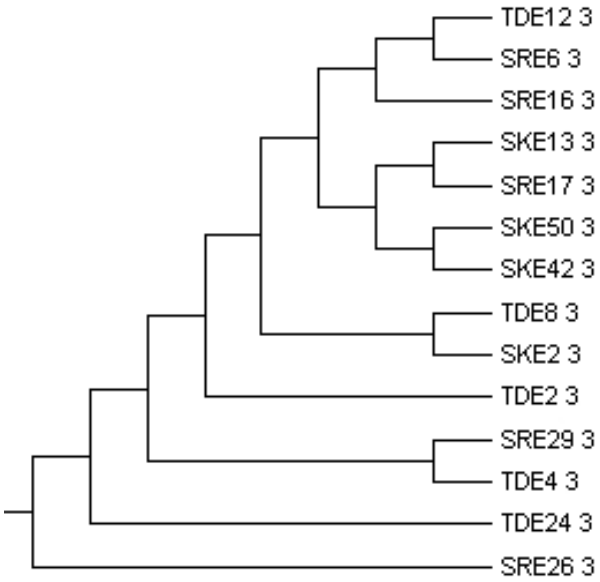
จำนวนชุดข้อมูลของการทำซ้ำ (No. of Replicates)	**กรณีทดสอบ (Test Case)**			**Running Time (Min.)**
	Manual	***Automatic Workflow***		
		No local service	***With local web services***	
1,000	-	✓	-	02.06
2,000	✓	-	-	15
2,000	-	✓	-	ล้มเหลว
2,000	-	-	✓	2.24
20,000	✓	-	-	ยากหรือเป็นไปไม่ได้ที่จะทดสอบ
20,000	-	-	✓	15.08
50,000	-	-	✓	30.17

รูปที่ 3.28 และรูปที่ 3.29 คือโครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์และกรดอะมิโนตามลำดับ ซึ่งได้รับการตรวจสอบความถูกต้องจากศูนย์วิจัยจีโนมิกส์และชีว-สารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์ เป็นความสัมพันธ์ทางระบบวิวัฒนาการของกุ้งจากแหล่งที่อยู่อาศัยต่างๆกัน 3 แหล่ง ความรู้ที่ได้จะสามารถช่วยคัดเลือกสายพันธุ์ในการศึกษาด้านอื่นๆต่อไปในอนาคต

โครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์ในรูปที่ 3.28 แสดงความสัมพันธ์ทางด้านพันธุกรรม อย่างเช่น ลำดับTDE24 และลำดับ SKE13 อยู่ในกลุ่มประเภท (Species) ด้วยกันและมีความสัมพันธ์ใกล้ชิดกับลำดับ SRE17 มากกว่าลำดับ SRE16 เป็นต้น โครงสร้างต้นไม้สายวิวัฒนาการของลำดับกรดอะมิโนในรูปที่ 3.29 แสดงให้เห็นว่าลำดับ SKE50 และลำดับ SKE42 อยู่ในกลุ่มประเภทเดียวกันและต่างก็สืบสายพันธุ์มาจาก TDE8 เป็นต้น



รูปที่ 3.28 โครงสร้างต้นไม้สายวิวัฒนาการของลำดับนิวคลีโอไทด์ของกุ้ง



รูปที่ 3.29 โครงสร้างต้นไม้สายวิวัฒนาการของลำดับกรดอะมิโนของกุ้ง

3.5 อภิปรายผลการทดลอง

วิธีการทำงานแบบเก่าหรือคัดลอกแล้ววางสามารถทำงานกับจำนวน 2,000 Bootstrapping replicates ได้ โดยนักชีวสารสนเทศที่มีความชำนาญในระบบงานและการใช้โปรแกรมแบบออฟไลน์กับเว็บแอ็พพลิเคชัน ใช้เวลาในการทำงานประมาณ 15 นาที ในขณะที่วิธีการแบบเวิร์คโฟลว์ที่นำเสนอในวิทยานิพนธ์นี้ เป็นการเรียกใช้เว็บเซอร์วิสต่างๆที่กระจายกันอยู่อินเตอร์เน็ต พบว่าที่จำนวน 2,000 Bootstrapping replicates นั้น เวิร์คโฟลว์ทำงานไม่สำเร็จเนื่องจากเกิดปัญหาการเกินเวลาของ Data flow ระหว่างเว็บเซอร์วิสดังที่ได้กล่าวแล้ว

สำหรับเวิร์คโฟลว์ใหม่ที่ใช้บริการท้องถิ่นที่พัฒนาขึ้นเอง สามารถแก้ปัญหาการเกินเวลาได้ และสามารถทำงานในการทำนายโครงสร้างต้นไม้สายวิวัฒนาการด้วยจำนวน 2,000 Bootstrapping replicates โดยเวลาเพียง 2.24 นาที ซึ่งเป็นเวลาที่ลดลงอย่างมีนัยสำคัญ และโครงสร้างต้นไม้สายวิวัฒนาการที่ได้มีความถูกต้อง โดยได้รับการตรวจสอบจากศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์

3.6 สรุป

ในบทนี้ได้กล่าวถึงการวาดและการเลือกบริการที่เหมาะสม สำหรับการทำงานด้วยวิธีการแบบเวิร์คโฟลว์ โดยให้ข้อเสนอแนะที่เป็นรูปธรรม และใช้เวิร์คโฟลว์ในการวิเคราะห์สนิปในกุ้ง ที่เป็นงานวิจัยร่วมกับศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์ มหาวิทยาลัยสงขลานครินทร์ เป็นกรณีศึกษา

วิทยานิพนธ์นี้นำเสนอการออกแบบ การพัฒนา และการทดสอบเวิร์คโฟลว์และเว็บเซอร์วิสสำหรับการวิเคราะห์สนิป และการสร้างโครงสร้างต้นไม้สายวิวัฒนาการในกุ้งแซบ๊วย ด้วยการใช้เทคโนโลยีมายกริดและทาเวอร์นา บริการท้องถิ่นที่สร้างขึ้นสามารถแก้ไขปัญหาการเกินเวลาและความไม่คงเส้นคงวาในการทำงานได้เป็นอย่างดี และสามารถลดเวลาการทำงานลงได้อย่างมีนัยสำคัญเมื่อเปรียบเทียบกับการทำงานด้วยวิธีการแบบเดิมหรือคัดลอกแล้ววาง เวิร์คโฟลว์ที่พัฒนาโดยใช้บริการท้องถิ่นเผยให้เห็นความสัมพันธ์ทางพันธุกรรมของกุ้ง นอกจากนี้ยังสามารถใช้เวิร์คโฟลว์วิเคราะห์ความสัมพันธ์ทางพันธุกรรมกับสิ่งมีชีวิตอื่นๆได้ ตัวอย่างเช่น สัตว์ทะเลประเภทหอยและหมึก (Molluse) และประเภทแมลง (Arthropod) โดยผู้ใช้งานสามารถป้อนลำดับนิวคลีโอไทด์ของสิ่งมีชีวิตเหล่านี้ให้กับเวิร์คโฟลว์ได้ โดยไม่ต้องปรับแต่งเวิร์คโฟลว์ใดๆเพิ่มเติมอีก

บทที่ 4

การออกแบบและการพัฒนาเวิร์คโฟลว์สำหรับการวิเคราะห์สนิปของมนุษย์

4.1 ที่มาและความสำคัญ

การวิเคราะห์หรือทำนายผลกระทบ อันเนื่องมาจากการเปลี่ยนแปลงในระดับเบสบนจีโนมมนุษย์หรือสนิป ซึ่งกระจายตัวอยู่ประมาณ 10 ล้านตำแหน่งบนโครโมโซม 23 คู่กับการตอบสนองต่อยาหรือภาวะการเป็นโรค จำเป็นต้องใช้ข้อมูลมากมายจากหลากหลายแห่งมาประกอบกันเป็นเวิร์คโฟลว์ และรวบรวมเป็นองค์ความรู้ใหม่เพื่อนำไปใช้ในการศึกษาถึงผลกระทบต่างๆ ของสนิปต่อไป โดยจะใช้เวิร์คโฟลว์นี้เป็นกรณีศึกษาของงานวิทยานิพนธ์

ในบทนี้จะกล่าวถึงประโยชน์ในด้านการพัฒนาเวิร์คโฟลว์ ขั้นตอนการเลือกบริการที่เหมาะสม การออกแบบระบบงานในงานวิจัยด้านสนิปของมนุษย์ การสร้างเวิร์คโฟลว์ของแต่กระบวนการ การรวบรวมทุกกระบวนการเป็นเวิร์คโฟลว์เดียวกัน การทดลองวัดเวลาการทำงานของเวิร์คโฟลว์และสรุป

4.2 ประโยชน์ในด้านการพัฒนาเวิร์คโฟลว์

ประโยชน์ในด้านการพัฒนาเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ ในงานวิจัยเภสัชพันธุศาสตร์มีดังนี้

- รวบรวมข้อมูลทางชีวสารสนเทศที่น่าสนใจ และเป็นประโยชน์จากหลายแหล่งข้อมูล โดยสร้างเป็นเวิร์คโฟลว์ในสิ่งแวดล้อมของโปรแกรมทาเวอร์นา
- รวบรวมผลการทำงานจากเว็บเซอร์วิสต่างๆ ที่มีความสามารถในการทำนายผลกระทบของการเปลี่ยนแปลงสนิป ที่อาจส่งผลถึงการทำงานของโปรตีนโดยสร้างเป็นเวิร์คโฟลว์ในสิ่งแวดล้อมของโปรแกรมทาเวอร์นา

- นำเทคโนโลยีด้านชีวสารสนเทศมาประยุกต์ใช้กับการเลือก หรือจัดลำดับความสำคัญของสนิปที่ควรศึกษาก่อน และอธิบายผลที่เกิดขึ้นจากความสัมพันธ์ระหว่างสนิปกับกลไกการเป็นโรค หรือการแพ้ยาชนิดต่างๆ ซึ่งในปัจจุบันยังไม่มีเว็บเซอร์วิสใดเพียงแห่งเดียวที่สามารถวิเคราะห์หรือทำนายผลกระทบอันเนื่องมาจากการเปลี่ยนแปลงของสนิปได้
- ช่วยประหยัดเวลาในการเข้าไปใช้งานเว็บเซอร์วิสต่างๆทีละบริการเพราะข้อมูลที่ได้มาจากการศึกษาสนิปแบบสแกนหาจีโนมทั้งหมด (Whole genome scan) จะมีปริมาณมาก หากทำแบบวิธีเก่าหรือคัดลอกแล้ววาง (Copying and Pasting) จะทำได้ยากมาก
- เป็นตัวอย่างของการนำข้อมูลจีโนมมากมายจากหลายแหล่ง มาประยุกต์ใช้ผ่านกระบวนการชีวสารสนเทศร่วมกับโปรแกรมทาเวอร์นา และเวิร์คโฟลว์ที่พัฒนาขึ้นจะเป็นไปเพื่อใช้ประโยชน์ทางการแพทย์ สามารถนำไปสร้างเป็นชุดตรวจกรองการเกิดโรคหรือการแพ้ยาต่างๆได้ในอนาคต
- ก่อให้เกิดองค์ความรู้ด้านความสัมพันธ์ระหว่างสนิป กับการเป็นโรคหรือการแพ้ยาชนิดต่างๆ รวมถึงผลกระทบของการเปลี่ยนแปลงสนิปที่อาจส่งผลถึงการทำงานของโปรตีน ซึ่งสามารถนำเวิร์คโฟลว์ที่ได้สร้างนี้ ไปประยุกต์ใช้กับการวิจัยอื่นๆที่เกี่ยวข้องกับสนิปจำนวนมาก
- สร้างระบบการรวบรวมข้อมูล ที่ออกแบบเฉพาะสำหรับการวิจัยด้านสนิปกับการเป็นโรคหรือการแพ้ยาชนิดต่างๆ รวมถึงผลกระทบของการเปลี่ยนแปลงสนิปที่อาจส่งผลถึงการทำงานของโปรตีนเป็นระบบแรกของประเทศไทย
- เกิดองค์ความรู้ใหม่ในการใช้เทคโนโลยีชีวสารสนเทศ จากโปรแกรมทาเวอร์-นาในการเชื่อมโยงเว็บเซอร์วิสต่างๆที่สำคัญ และมีข้อมูลปริมาณมากเพื่อให้มีความสามารถในการคำนวณหรือจัดลำดับความสำคัญของสนิป และสามารถวิเคราะห์หรือทำนายผลกระทบอันเนื่องมาจากการเปลี่ยนแปลงของสนิป
- เป็นก้าวสำคัญในการประยุกต์ใช้เทคโนโลยีทางคอมพิวเตอร์ มาแก้ปัญหาทางด้านวิทยาศาสตร์การแพทย์ ทำให้เกิดความร่วมมือกันระหว่างนักวิจัยด้านวิทยาศาสตร์คอมพิวเตอร์และวิทยาศาสตร์ชีวภาพ ในที่นี้คือมหาวิทยาลัยสงขลานครินทร์และมหาวิทยาลัยมหิดล

4.3 ขั้นตอนการค้นหาบริการที่เหมาะสม

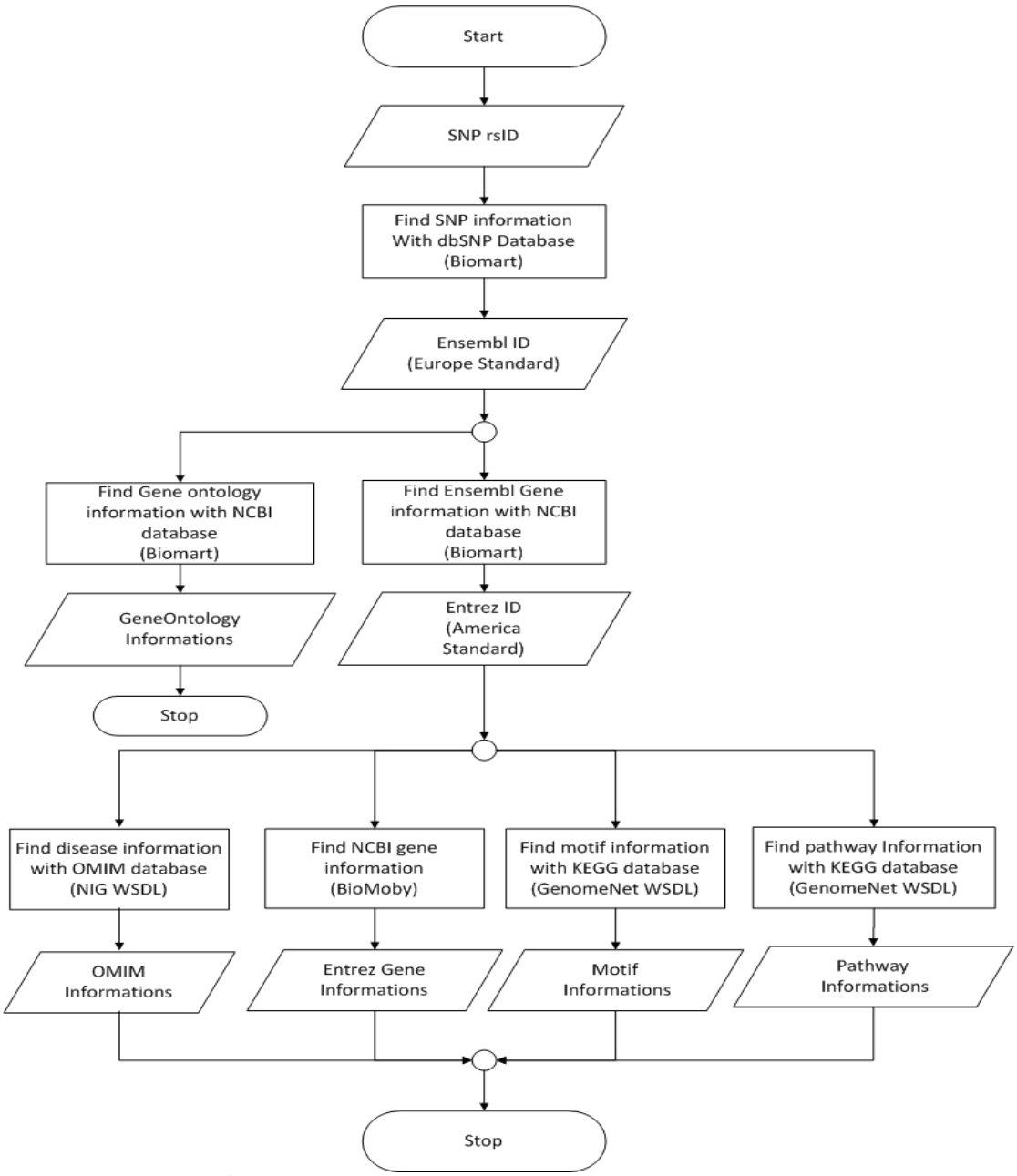
การเลือกบริการที่เหมาะสม ตรงกับความต้องการในการสร้างเวิร์คโฟลว์ได้เสนอไว้แล้วในวิทยานิพนธ์นี้ในบทที่ 3 ซึ่งในบทนี้สามารถใช้วิธีเดียวกันได้

4.4 การออกแบบระบบงานในงานวิจัยด้านสนิปของมนุษย์

กระบวนการค้นหาข้อมูลในงานวิจัยด้านสนิปของมนุษย์ ในงานวิทยานิพนธ์นี้เป็นการค้นหาข้อมูลจากฐานข้อมูลต่างๆที่กระจายกันบนอินเตอร์เน็ต และนำเป็นพื้นฐานไปสู่การพัฒนาเวิร์คโฟลว์โดยในบทนี้จะกล่าวถึงกระบวนการค้นหาข้อมูล Data flow diagram และการสร้างเวิร์คโฟลว์ดังนี้

4.4.1 กระบวนการค้นหาข้อมูล

การค้นหาข้อมูลของสนิปในงานวิจัยด้านเภสัชพันธุศาสตร์ สามารถออกแบบการทำงานในภาพรวมในรูปแบบ Flowchart ได้ดังรูปที่ 4.1 ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 4.1 Flowchart การทำงานในภาพรวมของเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

1) การค้นหาข้อมูลยีนที่เกี่ยวข้องกับสนิปที่สนใจจากฐานข้อมูล dbSNP ของ NCBI โดยใช้ SNP ID จำนวน 500 – 1,000 IDs จากห้องปฏิบัติการจริงซึ่งได้จากการวิเคราะห์ทางหลักสถิติจากจำนวนประมาณ 550,000 ตำแหน่ง ผลลัพธ์ของกระบวนนี้คือ SNP Information และข้อมูลที่จะนำไปประมวลต่อคือ Ensembl ID [50]

2) การค้นหารายละเอียดเกี่ยวกับยีนที่สนใจ จากฐานข้อมูล Homo sapiens genes (NCBI36) ของ NCBI [51][52] เช่น รหัสและชื่อของยีน ผลลัพธ์ของกระบวนการนี้คือ Ensembl Gene Information และข้อมูลที่จะไปประมวลต่อคือ Entrez ID [53]

3) การค้นหาข้อมูล Gene Ontology ที่เกี่ยวข้องกับยีนที่สนใจ จากฐานข้อมูล Homo sapiens genes (NCBI36) ของ NCBI ผลลัพธ์ของกระบวนการนี้คือ Gene Ontology Information

4) การค้นหาข้อมูลของโรคต่างๆที่เกี่ยวข้องกับยีนที่สนใจ หรือ Online Mendelian Inheritance in Man (OMIM) [54] โดยใช้อินพุตคือ Entrez ID จากฐานข้อมูลของ DDBJ และผลลัพธ์ของกระบวนการนี้คือ OMIM Information

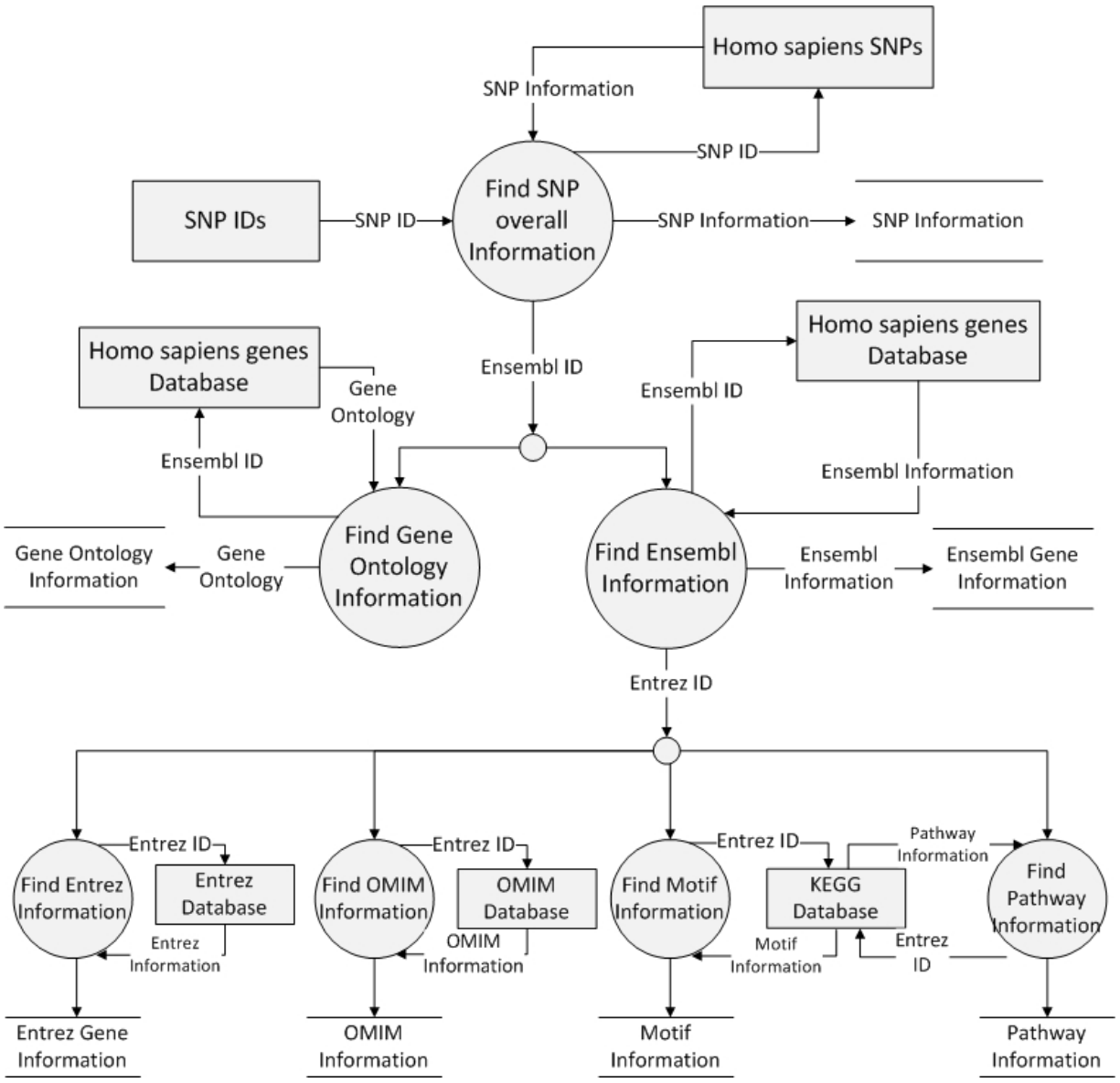
5) การค้นหาข้อมูลเกี่ยวกับ Motif [55] ของยีนที่เกี่ยวข้องกับยีนที่สนใจ โดยใช้อินพุตคือ Entrez ID จากฐานข้อมูล Kyoto Encyclopedia of Gene and Genomes (KEGG) [56] และผลลัพธ์ของกระบวนการนี้คือ Motif Information

6) การค้นหาข้อมูลเกี่ยวกับ Pathway หรือกลไกการทำงานของยีนที่สนใจ [57] โดยใช้อินพุตคือ Entrez ID จากฐานข้อมูล KEGG ซึ่งจะแสดงกลไกการทำงานของยีนในรูปแบบของภาพกราฟิกที่เข้าใจง่าย และผลลัพธ์ของกระบวนการนี้คือ Pathway Information

7) การค้นหาชื่ออื่นๆของยีนที่สนใจ ที่มีความเกี่ยวข้องหรือใกล้เคียงกันจากฐานข้อมูล NCBI และผลลัพธ์ของกระบวนการนี้คือ Entrez Gene Information

4.4.2 Data flow diagram ของเวิร์คโฟลว์

จากกระบวนการค้นหาข้อมูลจากฐานข้อมูลที่สนใจ ในที่ต่างๆที่กระจัดกระจายกันอยู่บนเครือข่ายอินเตอร์เน็ต สามารถอธิบายการไหลและกระบวนการของข้อมูลด้วย Data flow diagram ได้ดังรูปที่ 4.2 ซึ่งจะเห็นว่าอินพุตที่ใช้ในเวิร์คโฟลว์จะเริ่มต้นด้วย SNP IDs ซึ่งผู้ใช้งานเวิร์คโฟลว์เป็นผู้ระบุ ต่อจากนั้นเวิร์คโฟลว์จะเริ่มทำงานไปตามกระบวนการที่ออกแบบไว้ ซึ่งจะได้เอาท์พุตสำคัญที่ถูกนำไปใช้งานเป็นอินพุตสำหรับกระบวนการถัดไปคือ จาก SNP IDs ได้เป็น Ensembl IDs และจาก Ensembl IDs ได้เป็น Entrez IDs

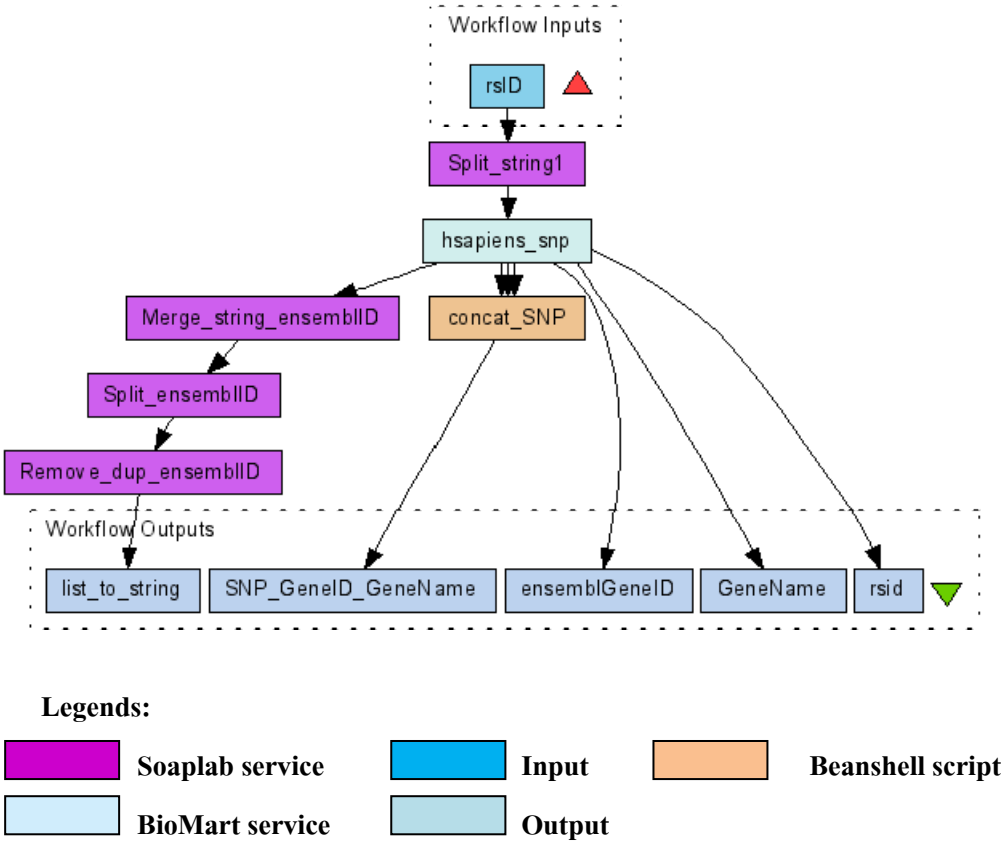


รูปที่ 4.2 Data flow diagram ของเวิร์คโฟลว์

4.5 การสร้างเวิร์คโฟลว์ของแต่กระบวนการ (Implementation)

จากการออกแบบ Data flow ดังรูปที่ 4.2 สามารถนำแต่ละกระบวนการมาสร้างเป็นเวิร์คโฟลว์ในงานวิจัยด้านสนิปด้วยโปรแกรมทาเวอร์นาได้ดังนี้

1) การค้นหาข้อมูลสนิปของมนุษย์ซึ่งใช้ SNP IDs เป็นอินพุตค้นหาจากฐานข้อมูล dbSNP โดยใช้เว็บเซอร์วิสไบโอมาร์ทชื่อ *hsapiens_snp* (Homo sapiens SNPs) เอาท์พุตที่ต้องการคือ SNP Gene ID, Ensembl Gene ID, Gene Name และ rs ID หรือ SNP ID โดยข้อมูลที่จะนำไปเป็นอินพุตให้กับกระบวนการถัดไปคือ Ensembl Gene ID และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.3 โดยมีบริการหลักคือ *hsapient_snp* ทำหน้าที่ค้นหาข้อมูลสนิป แต่ผลลัพธ์การค้นหานั้น ยังไม่อยู่ในรูปแบบที่สามารถนำไปใช้เป็นอินพุตกับกระบวนการถัดไปได้ทันที จึงต้องปรับแต่งผลลัพธ์ด้วยบริการท้องถิ่นของ Java (Java loca service) เช่น บริการชื่อ *Merge_string_ensemblID* และยังมีการใช้งานบริการท้องถิ่นบีนเชลล์ ชื่อ *concat_SNP* ด้วย ตัวอย่างดังตารางที่ 4.1

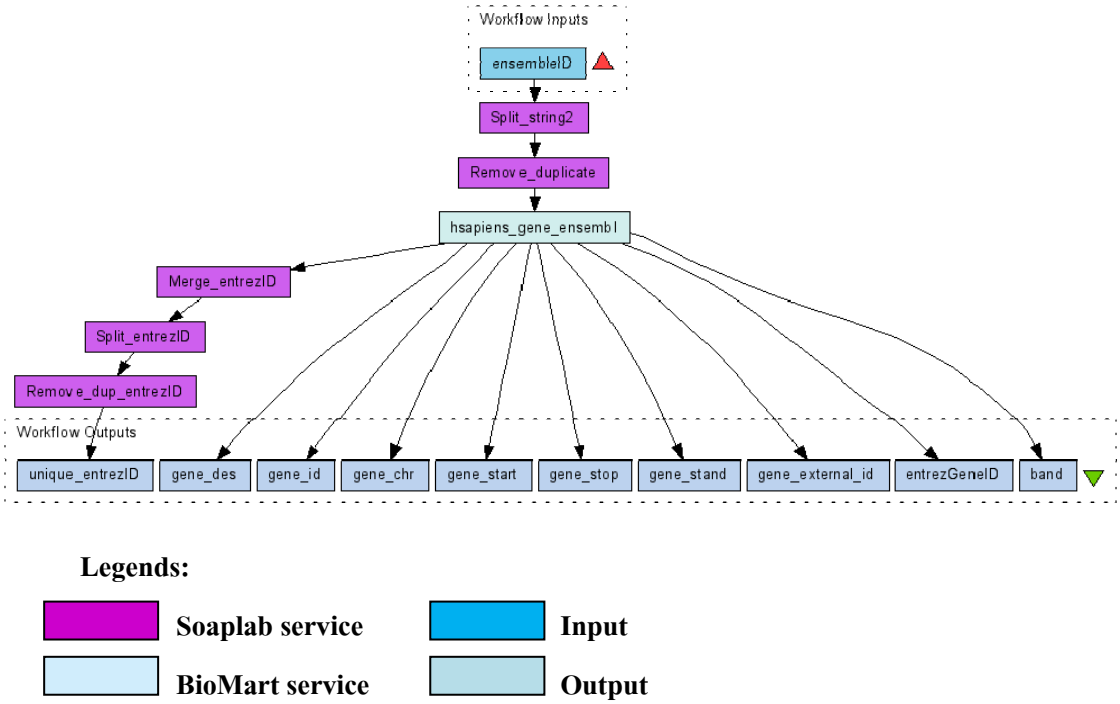


รูปที่ 4.3 เวิร์คโฟลว์การค้นหาข้อมูลสนิปที่สนใจ

ตารางที่ 4.1 ตัวอย่างการจัดการอินพุตและเอาท์พุตในเวิร์คโฟลว์

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Split_string1 Split_ensemblID	แยก string	“rs1043502 rs10799658”	“rs1043502” “rs10799658”
Merge_string_ensemblID	รวมหลาย string เป็น string เดียว	“ENSG00000158828” “ENSG00000117242” “ENSG00000117242”	“ENSG00000158828 ENSG00000117242 ENSG00000117242”
Remove_dup_ensemblID	ลบ string ที่ซ้ำซ้อน	“ENSG00000158828” “ENSG00000117242” “ENSG00000117242”	“ENSG00000158828” “ENSG00000117242”
Concat_SNP	รวบรวมเอาท์พุตเข้าด้วยกัน	rs1043502 PINK1 ENSG00000158828	“rs1043502,PINK1,ENSG00000158828”

2) การค้นหาข้อมูลของยีนจากฐานข้อมูล Homo sapiens genes (NCBI36) โดยใช้บริการชื่อ *hsapient_gene_ensembl* โดยมีอินพุตคือ Ensembl Gene ID ที่ได้เวิร์คโฟลว์ในรูปที่ 4.3 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.4 ซึ่งมีเอาท์พุตดังนี้ Unique Entrez ID, Gene Description, Gene ID, Gene Chromosome, Gene Start Position, Gene Stop Position, Gene Stand, Gene External ID (Gene Name) และ Gene Band. โดยข้อมูลที่จะนำไปเป็นอินพุตให้กับกระบวนการถัดไปคือ Entrez ID มีการใช้บริการท้องถิ่นของ Java เพื่อจัดการกับเอาท์พุตที่จะไปเป็นอินพุตให้กับกระบวนถัดไป ตัวอย่างดังตารางที่ 4.2



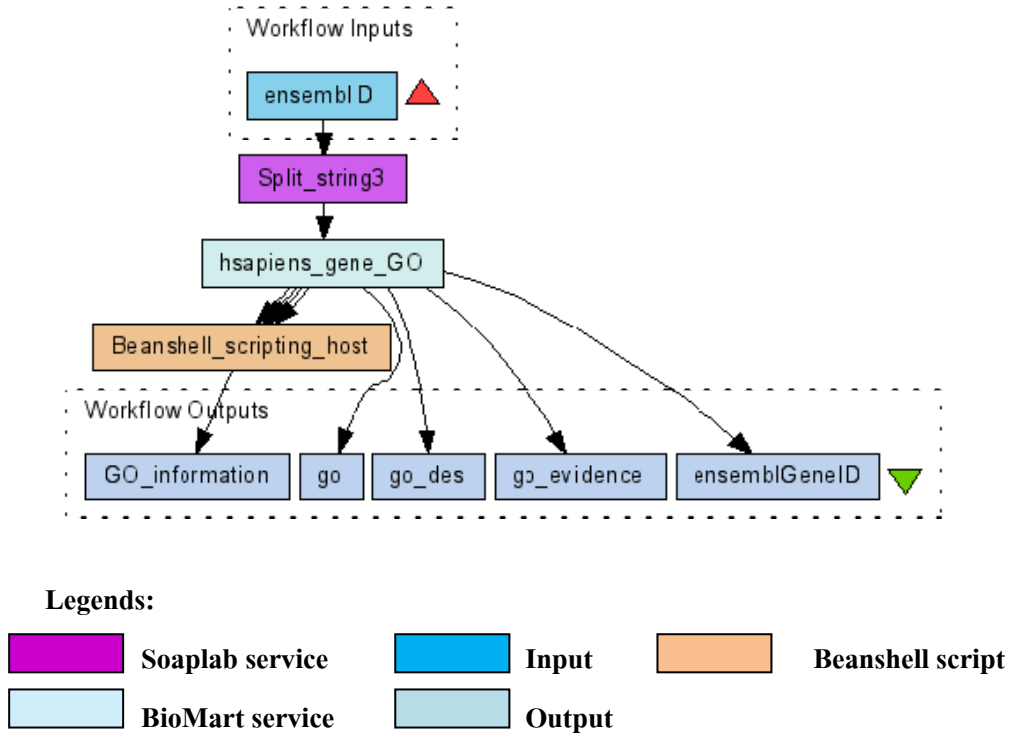
รูปที่ 4.4 เวิร์คโฟลว์การค้นหาข้อมูลของยีน

ตารางที่ 4.2 การทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสท้องถิ่นในการค้นหาข้อมูลยีน

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Spilt_string2	แยก string	“ENSG00000117242 ENSG00000158828”	“ENSG00000117242” “ENSG00000158828”
Remove_duplicate	ลบ string ที่ซ้ำซ้อน	“ENSG00000117242” “ENSG00000158828”	“ENSG00000117242” “ENSG00000158828”
Hsapiense_gene_ensembl	หาข้อมูลของยีน	“ENSG00000117242” “ENSG00000158828”	“1650” “65018”

3) การค้นหาข้อมูล Gene Ontology จากฐานข้อมูล Homo sapiens genes (NCBI36) โดยใช้บริการไบโอมาร์ทชื่อ *hsapient_gene_ensembl* ตั้งชื่อเพื่อสื่อความหมายในจุดประสงค์การทำงานว่า *hsapiens_gene_GO* โดยมีอินพุตคือ Ensembl Gene ID ที่ได้จากเวิร์คโฟลว์ในรูปที่ 4.3 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.5 ซึ่งมีเอาท์พุตดังนี้ Gene Ontology, Gene

Description, Gene Ontology Evidence และ Ensembl Gene ID โดยมีบริการท้องถิ่นบีนเชลล์รวบรวมเอาท์พุตต่างๆเหล่านี้เข้าด้วยกันเป็นรายงานดังตารางที่ 4.3

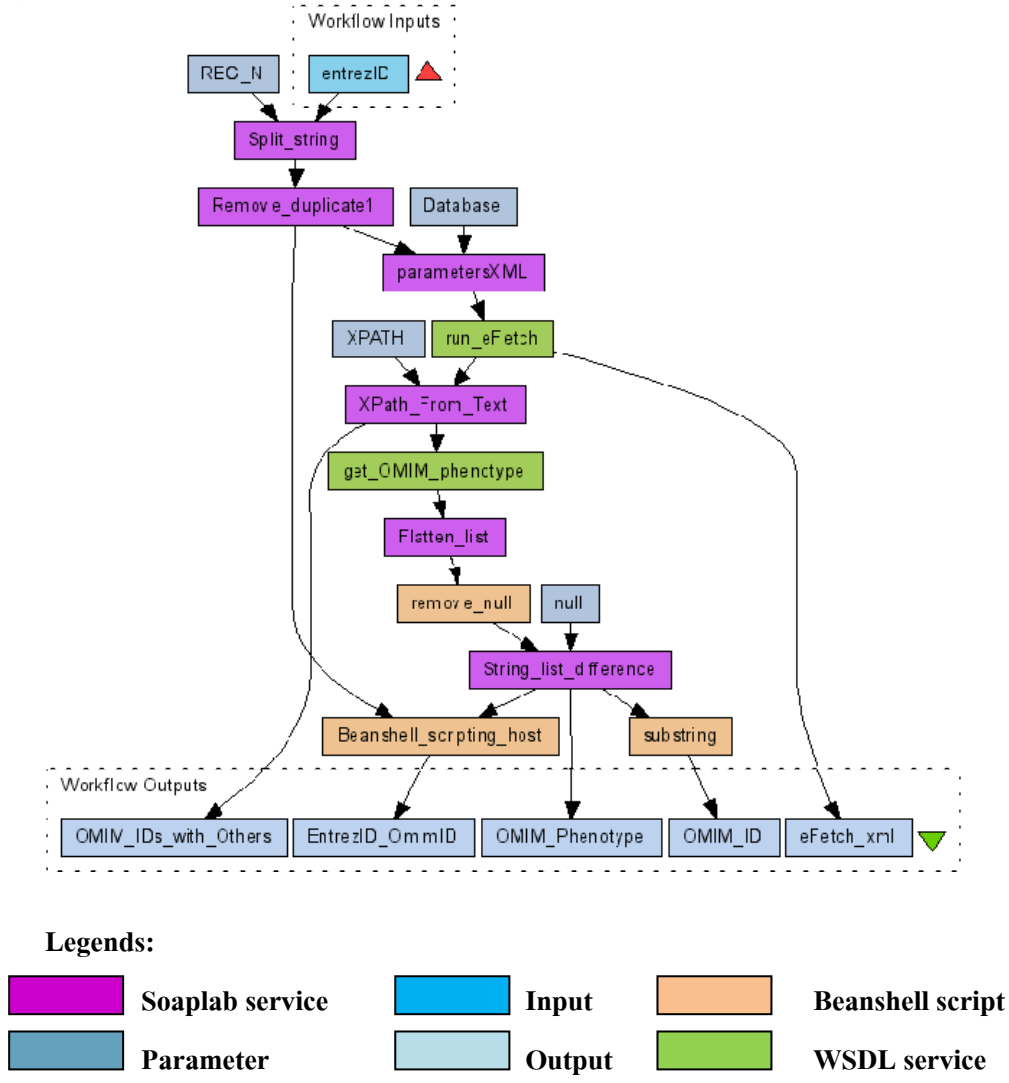


รูปที่ 4.5 เวิร์คโฟลว์การค้นหาข้อมูล Gene Ontology

ตารางที่ 4.3 การทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสท้องถิ่นในการค้นหาข้อมูล Gene Ontology

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Spilt_string3	แยก string	“ENSG00000117242 ENSG00000158828”	“ENSG00000117242” “ENSG00000158828”
Beanshell_scriptin g_host	บีนเชลล์สคริปต์สำหรับต่อ string เข้าด้วยกัน	gene_id, go, go_des, evi	GO_information = gene_id + "," + go + "," + go_des + "," + evi เช่น ENSG00000117242,GO:0016020,membrane,IEA

4) การค้นหาข้อมูล OMIM จากฐานข้อมูลของ DDBJ [40] โดยใช้เว็บเซอร์วิสชนิด WSDL ชื่อ *get_OMIM_phenotype* โดยมีอินพุตคือ Entrez Gene ID ที่ได้จากเวิร์คโฟลว์ในรูปที่ 4.3 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.6 โดยมีเอาท์พุตที่สำคัญคือ OMIM ID และ OMIM Phenotype อธิบายตัวอย่างที่สำคัญดังตารางที่ 4.4

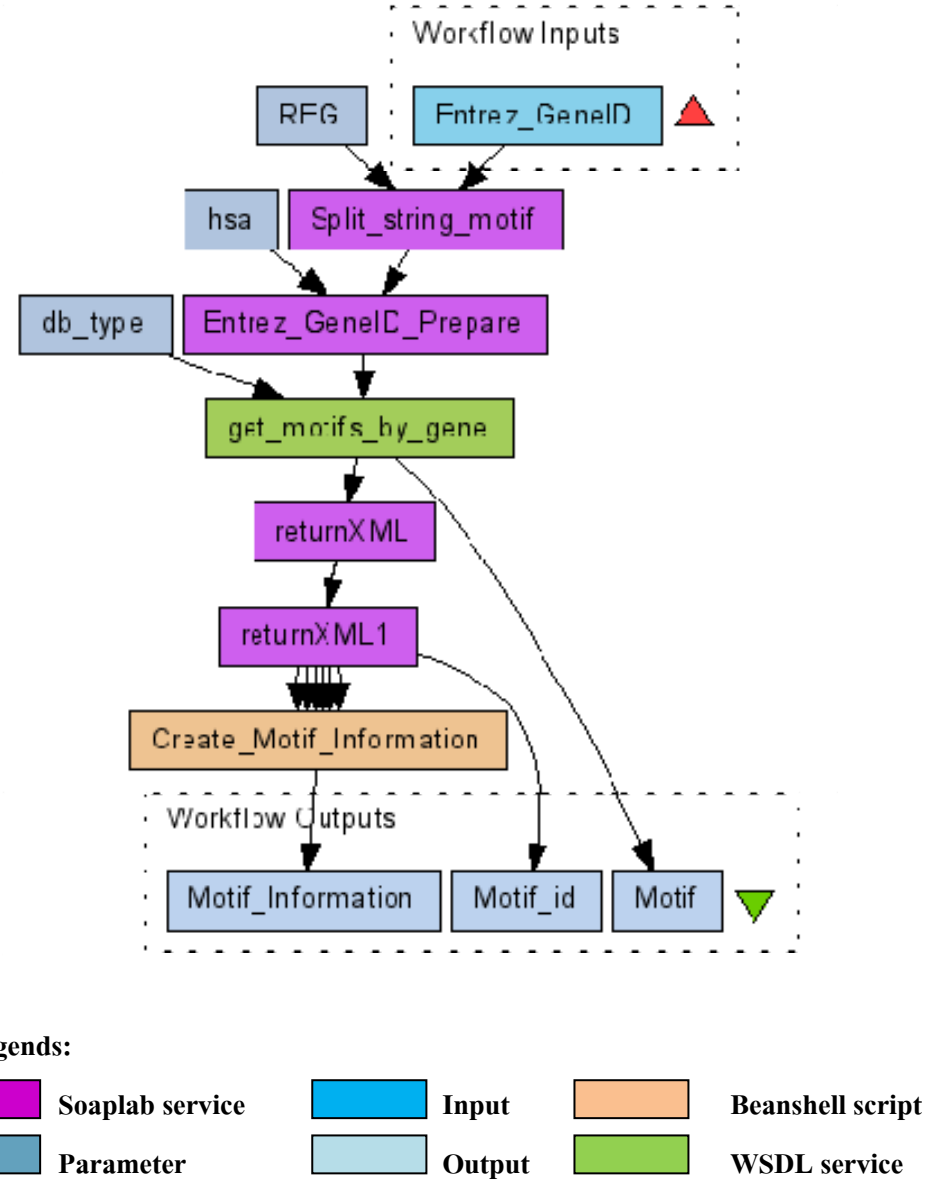


รูปที่ 4.6 เวิร์คโฟลว์การค้นหาข้อมูล OMIM

ตารางที่ 4.4 ตัวอย่างการทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสในการค้นหาข้อมูล OMIM

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Spilt_string3	แยก string	“1650 65018”	“1650” “65018”
Run_eFetch	ค้นหาข้อมูลของยีนจากฐานข้อมูล NCBI	“1650” “65018”	ข้อมูลยีนอยู่ในรูปแบบของไฟล์ XML ขนาดใหญ่
XPath_From_Text	XPath processor สำหรับดึงอิลิเมนท์ข้อมูลตามที่ระบุใน Expression	ข้อมูลยีนอยู่ในรูปแบบของไฟล์ XML ขนาดใหญ่ และ XPath Expression ในการดึง OMIM ID	“608309” “605909” “602544”
get_OMIM_phenotype	ค้นหาข้อมูล OMIM ของยีนโดยใช้ OMIM ID	“608309” “605909” “602544”	OMIM Phenotype แยกเป็นรายงการตาม OMIM ID

5) การค้นหาข้อมูล Motif ของยีนจากฐานข้อมูล KEGG โดยใช้เว็บเซอร์วิสชื่อ *get_motifs_by_gene* โดยมีอินพุตคือ Entrez Gene ID ที่ได้จากเวิร์คโฟลว์ในรูปที่ 4.4 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.7 และมีเอาท์พุตคือข้อมูลเกี่ยวกับ Motif ของ Entrez Gene ID ที่สนใจคือ Motif ID, Definition, Gene ID, Start Position, Endposition และ Score ตัวอย่างการทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสในการค้นหาข้อมูล Motif ดังตารางที่ 4.5

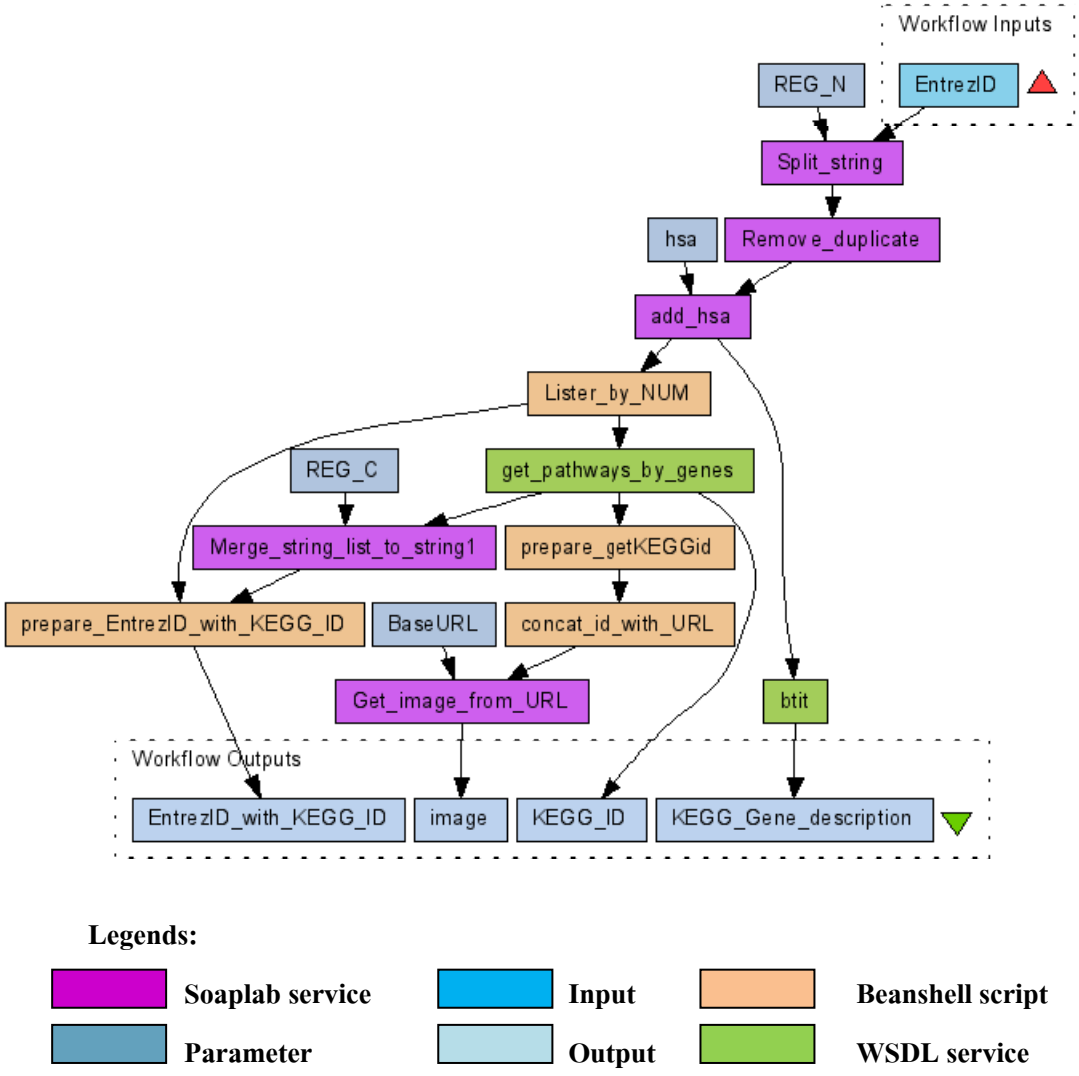


รูปที่ 4.7 เวิร์คโฟลว์การค้นหาข้อมูล Motif ของยีนจากฐานข้อมูล KEGG

ตารางที่ 4.5 ตัวอย่างการทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสในการค้นหาข้อมูล Motif

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Spilt_string_motif	แยก String	“1650 65018”	“1650” “65018”
Entrez_Gene ID_Prepare	เตรียม Entrez ID ให้เป็น Mofit ID โดยการต่อ ‘hsa:’ เข้าไปอยู่ข้างหน้า Entrez ID	“1650” “65018”	“hsa:1650” “hsa:65018”
returnXML returnXML1	ดึงอิลิเมนท์ของข้อมูลต่างๆจากเอกสาร XML ของข้อมูล Motif	เอกสาร XML ของข้อมูล Mofit	อิลิเมนท์ข้อมูลคือ motif_id, definition, genes_id, start_position, end_position , score
Create_Mofit_Information	บีนเชลล์สคริปต์ในการรวบรวมอิลิเมนท์เข้าด้วยกันเป็นรายงาน	motif_id, definition, genes_id, start_position, end_position , score	motif_info = motif_id +", " + definition + ", "+ genes_id + ", " + start_position + ", " + end_position + ", " + score;

6) การค้นหาข้อมูล Pathways หรือกลไกการทำงานของยีนจากฐานข้อมูล KEGG [8] โดยใช้เว็บเซอร์วิส WSDL ชื่อ *get_pathways_by_genes* โดยมีอินพุตคือ Entrez Gene ID ที่ได้จากเวิร์คโฟลว์ในรูปที่ 4.4 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.8 โดยมีเอาท์พุตที่สำคัญคือ ข้อมูล Pathways และรูปกราฟิกของ Pathways ตัวอย่างการทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสในการค้นหาข้อมูล Pathways ดังตารางที่ 4.6

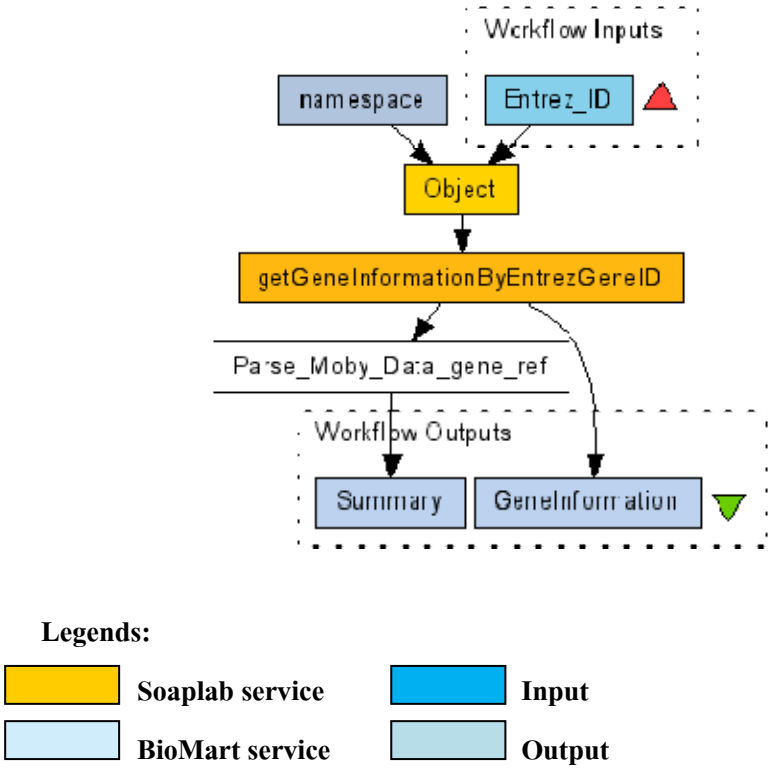


รูปที่ 4.8 เวิร์คโฟลว์การค้นหา Pathways หรือกลไกการทำงานของยีนจากฐานข้อมูล KEGG

ตารางที่ 4.6 ตัวอย่างการทำงานและข้อมูลอินพุตเอาท์พุตของเว็บเซอร์วิสในการค้นหาข้อมูล Pathways

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	ตัวอย่างเอาท์พุต
Spilt_string	แยก string	“1650 65018”	“1650” “65018”
add_hsa	เตรียม Entrez Gene ID ให้เป็น Mofit ID โดยการต่อ ‘hsa:’ เข้าไปอยู่ข้างหน้า Entrez Gene ID	“1650” “65018”	“hsa:1650” “hsa:65018”
Lister_by_NUM	เตรียมข้อมูลอินพุตให้เป็นรายการ (list) สำหรับบริการถัดไป	“hsa:1650” “hsa:65018”	“hsa:1650” “hsa:65018”
get_pathways_by_genes	ค้นหา Pathways ID จาก Mofit ID	“hsa:7167” “hsa:24”	“path:hsa00010,path:hsa00031,path:hsa00051,path:hsa00710” “path:hsa02010”
Concat_id_with_URL	บีนเชลล์สคริปต์สำหรับเชื่อม Pathway ID เข้ากับ Base URL เพื่อดึงข้อมูลภาพกราฟิกของ Pathway	Base URL และ Pathway ID คือ http://www.genome.jp + "/kegg/pathway/hsa/hsa" + id + ".gif";	เช่น http://www.genome.jp/kegg/pathway/hsa/hsa00010.gif
Get_image_from_URL	บริการท้องถิ่น Java สำหรับดึงข้อมูลภาพกราฟิกของ Pathway ด้วยคำร้องขอ GET ของ HTTP	http://www.genome.jp/kegg/pathway/hsa/hsa00010.gif	ภาพกราฟิกของ Pathway ID นั้นๆ

7) การค้นหาข้อมูลของยีนจากฐานข้อมูล Entrez ของ NCBI โดยใช้เว็บเซอร์วิส BioMoby ชื่อ *getGeneInformationByEntrezGeneID* โดยมีอินพุตคือ Entrez Gene ID ที่ได้จากเวิร์คโฟลว์ในรูปที่ 4.4 และสร้างเวิร์คโฟลว์ได้ดังรูปที่ 4.9 ได้เอาท์พุตคือ ข้อมูลเกี่ยวกับ Entrez Gene การใช้บริการ BioMoby จะต้องระบุ namespace ให้กับ Object ของอินพุต ในที่นี้คือ *EntrezGene_EntrezGeneID* เพื่อบอกว่าเป็น Entrez Gene ID และเอาท์พุตที่ได้ก็จะอยู่ในรูปแบบของ Object เช่นเดียวกัน ซึ่งบริการ BioMoby ก็ได้เตรียมตัวดึงข้อมูลเอาท์พุตต่างๆจาก Object เอาไว้ให้แล้วโดยผู้ใช้งานไม่จำเป็นต้องสร้างเว็บเซอร์วิสหรือบีนเชลล์สคริปต์เพื่อทำงานดังกล่าว



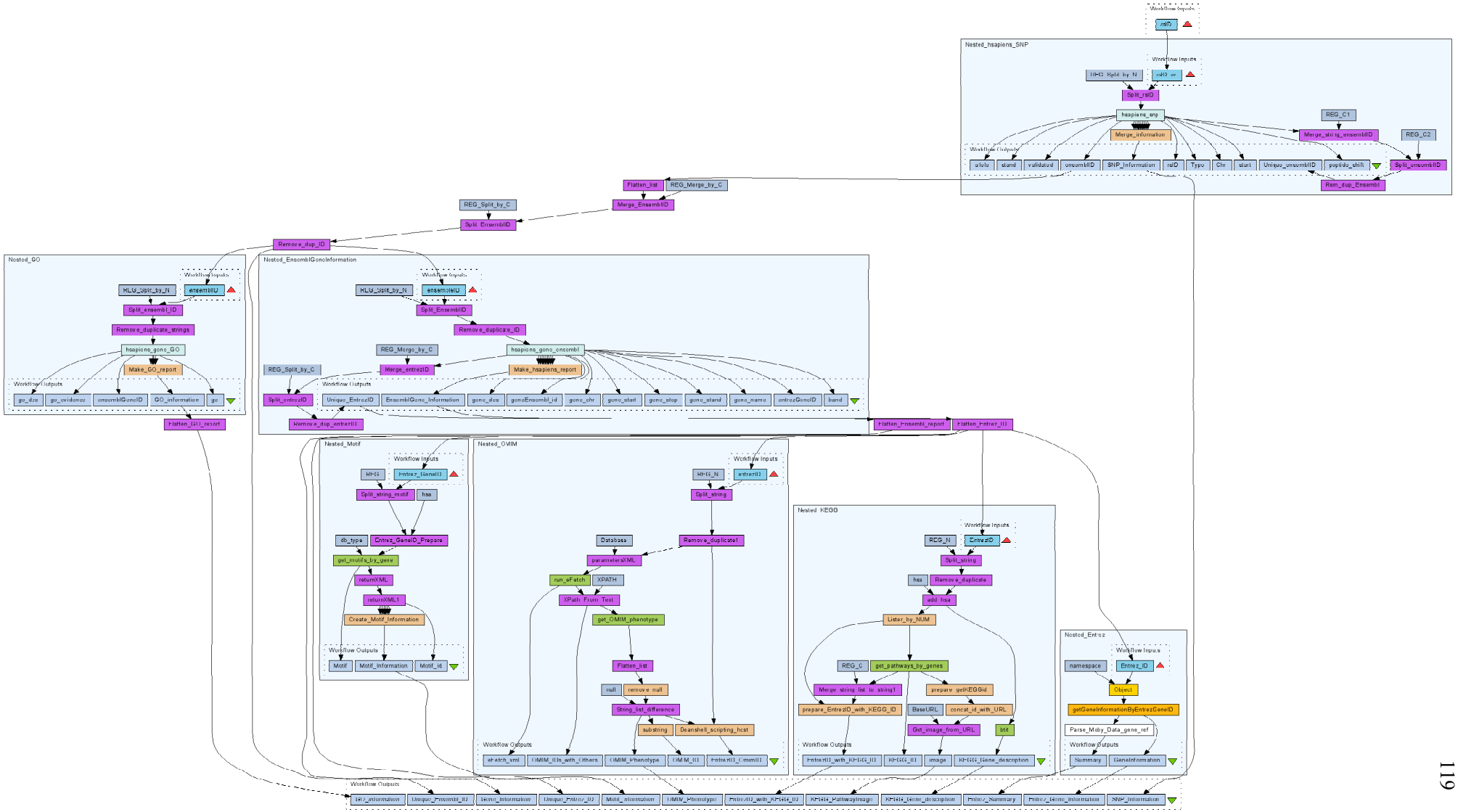
รูปที่ 4.9 เวิร์คโฟลว์การค้นหาข้อมูลของยีนจากฐานข้อมูล Entrez จาก NCBI

4.6 การรวบรวมทุกกระบวนการเป็นเวิร์คโฟลว์เดียวกัน

จากการสร้างและการทดสอบเวิร์คโฟลว์ของแต่ละกระบวนการแล้ว สามารถนำเวิร์คโฟลว์ของแต่ละกระบวนการมารวบรวมไว้ด้วยกัน และปรับแต่งอินพุตและเอาท์พุตตามความเหมาะสม เพราะเวิร์คโฟลว์ในแต่ละกระบวนการเมื่อรวบรวมเข้ามาด้วยกันแล้ว จะมีการทำงานเป็น

ลักษณะของลูปหรือ Nested workflow ดังนั้นเอาท์พุตต่างๆภายใน Nested workflow จึงต้องมีการต่อการเชื่อมต่อข้อมูล (Data link) ของเอาท์พุตที่สนใจและนำไปใช้งานต่อไป

ในเบื้องต้นได้ทำการเชื่อมต่อทุกเวิร์คโฟลว์ของกระบวนการที่เกี่ยวข้อง ดังที่ได้ทดลองสร้างตามข้อที่ 4.5 ดัง รูปที่ 4.10 โดยมีคุณสมบัติของเวิร์คโฟลว์ดังตารางที่ 4.7



รูปที่ 4.10 เวิร์คโฟลว์การค้นหาข้อมูลสนิป

จากตารางที่ 4.7 แสดงคุณสมบัติของเวิร์คโฟลว์โดยมีบริการทั้งหมด 55 บริการ เป็นบริการไอโบมาร์ทที่ทำหน้าที่คิวรีข้อมูล 3 บริการ คือ *hsapiens_snp*, *hsapiens_ensemb* และ *hsapaines_gene_GO* ซึ่งแต่ละบริการเหล่านี้ เป็นส่วนผลิตเอาท์พุตแล้วส่งต่อเอาท์พุตนี้เป็นอินพุตให้กับบริการอื่นๆอีกหลายบริการ ซึ่งประกอบด้วย 3 ฟิลเตอร์และ 22 แอ็ตตริบิวต์และมีดาตาลิงค์ทั้งหมด 161 ดาตาลิงค์

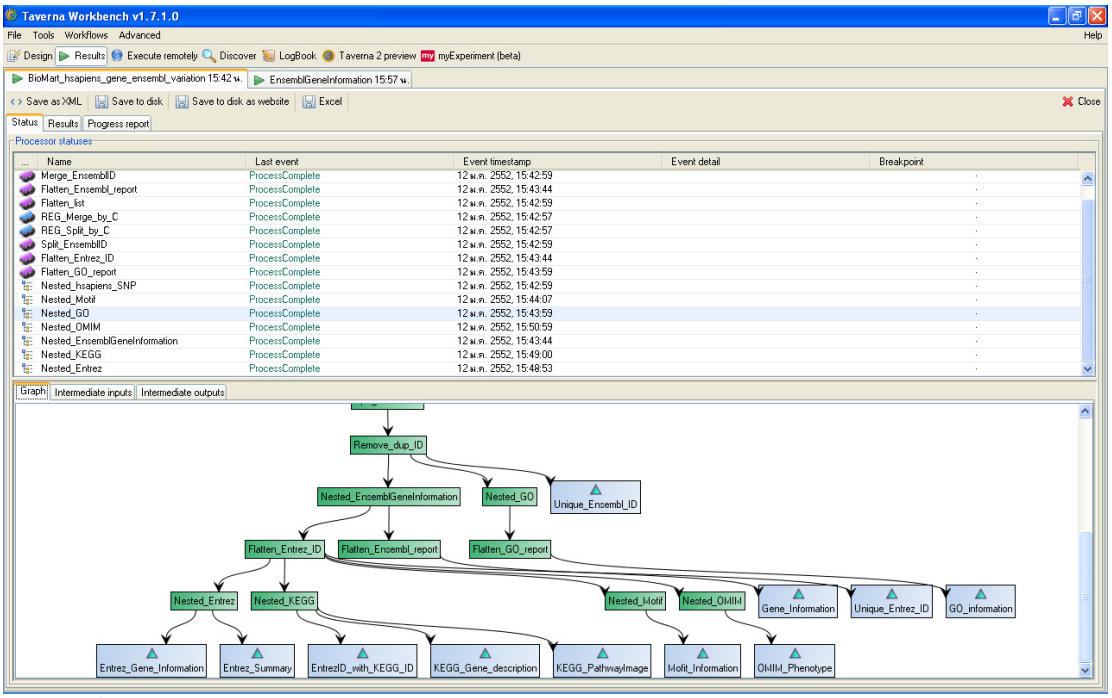
ตารางที่ 4.7 คุณสมบัติของเวิร์คโฟลว์ค้นหาข้อมูลสนิปและระยะเวลาในการทำงาน

รายการ	**จำนวน**	**หมายเหตุ**
บริการทั้งหมด	55	-
ดาตาลิงค์ทั้งหมด	161	-
บริการที่เกี่ยวเนื่องกับบริการไบโอมาร์ท	17	เป็นบริการไบโอมาร์ท 3 บริการ
ดาตาลิงค์ที่เกี่ยวเนื่องกับบริการไบโอมาร์ท	70	-
ฟิลเตอร์	3	เป็นฟิลเตอร์ของบริการไบโอมาร์ท 3 บริการ
แอ็ตตริบิวต์	22	เป็นแอ็ตตริบิวต์ของบริการไบโอมาร์ท 3 บริการ
ข้อมูลอินพุต	1,000	SNP IDs
เวลาทำงาน	14.00.50	ชั่วโมง
ผลการทำงาน	ทุกบริการทำงานสำเร็จ	ยกเว้นส่วนการค้นหา Gene Ontology

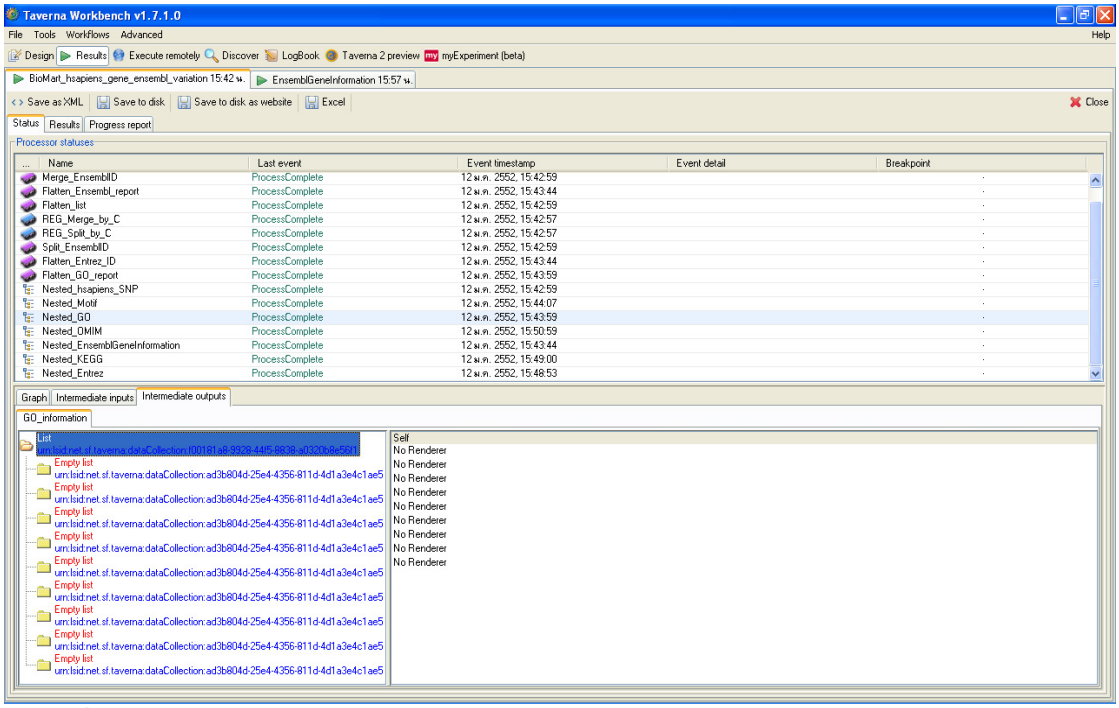
4.7 การทดลองวัดเวลาการทำงานของเวิร์คโฟลว์

การทดลองวัดเวลาการทำงานของเวิร์คโฟลว์โดยใช้ SNP ID จำนวน 1,000 IDs เป็นข้อมูลอินพุต ปรากฏว่าเวิร์คโฟลว์ใช้เวลาประมาณ 14 ชั่วโมง ทุกบริการสามารถทำงานสำเร็จยกเว้นในส่วนของการค้นหา Gene ontology ซึ่งโปรแกรมทาเวอร์นาระบุว่าทำงานได้สำเร็จเช่นเดียวกันโดยไม่มีรายงานข้อผิดพลาดแต่ไม่ได้ผลิตผลลัพธ์ใดๆดังรูปที่ 4.11 และรูปที่ 4.12 ตามลำดับ ซึ่งจริงๆแล้วคือทำงานไม่สำเร็จ

ในกรณีกระบวนการทำงานตามปกติของเวิร์คโฟลว์นี้ ป้อนอินพุตให้เวิร์คโฟลว์ทำงานเป็น SNP ID จำนวน 1,000 IDs ตามที่ได้ออกแบบไว้ ซึ่งกว่าจะทราบว่าบางส่วนของเวิร์ค-โฟลว์ทำงานไม่ถูกต้องตรงตามความต้องการก็ต้องรอถึง 14 ชั่วโมง คือรายการที่ไม่ข้อมูล (Empty list) ก็ต้องเสียเวลารอจนกว่าเวิร์คโฟลว์จะทำงานเสร็จ และเมื่อทำงานเสร็จแล้วก็ไม่สามารถที่จะทราบปัญหาได้ทันทีว่าปัญหาเกิดจากสาเหตุอะไร เพราะทุกบริการแจ้งสถานการณ์ทำงานถูกต้อง ซึ่งถัดจากนี้ ก็ต้องตรวจสอบเวิร์คโฟลว์เพื่อหาสาเหตุของข้อผิดพลาด และโดยเฉพาะอย่างยิ่งการผลิตผลลัพธ์เป็นรายการที่ไม่มีข้อมูล (Empty list) จะตรวจสอบได้ยาก เนื่องจากบางกรณีข้อมูลชีว-สารสนเทศก็ให้ผลลัพธ์เป็นรายการที่ไม่มีข้อมูลได้ เช่น Ensembl ID หนึ่งๆ ไม่จำเป็นว่าจะต้องมีข้อมูล Entrez ID ด้วยเสมอไป หรือ Entrez ID หนึ่งๆ ก็ไม่จำเป็นว่าจะต้องมีข้อมูล Ensembl ID ด้วยเช่นกัน เนื่องจาก Ensembl เป็นมาตรฐานของยุโรป แต่ Entrez เป็นมาตรฐานของอเมริกา ดังนั้นบางช่วงเวลาระยะหนึ่งๆในการปรับปรุงข้อมูลของแต่ละมาตรฐาน จะทำให้ไม่ได้ข้อมูลที่สอดคล้องกันดังกล่าว ซึ่งจะต้องรอจนกว่าการจัดการการปรับปรุงนั้นเสร็จสิ้นสมบูรณ์ จึงจะได้ข้อมูลที่ถูกต้องและสอดคล้องเป็นหนึ่งเดียวกัน



รูปที่ 4.11 หน้าต่างแสดงทุกบริการในเวิร์คโฟลว์ทำงานสำเร็จโดยมีแสดงด้วยบล็อกสีเขียว



รูปที่ 4.12 หน้าต่างแสดงส่วนการค้น Gene Ontology ทำงานได้สำเร็จแต่ไม่ได้ให้ผลลัพธ์ใดๆ

4.8 สรุป

ในบทนี้ได้กล่าวถึงการพัฒนาวิธีการทำนายผลกระทบ อันเนื่องมาจากการเปลี่ยนแปลงสนิปต่างๆ รวมทั้งสร้างระบบที่มีความสามารถในการคัดเลือกตำแหน่งสนิปที่น่าสนใจในการนำไปศึกษาต่อทางด้านหน้าที่การทำงานต่อไปด้วยเวิร์คโฟลว์ การแก้ปัญหางานวิจัยใช้โปรแกรมทาเวอร์นาที่มีความสามารถในการสร้างและจัดการรูปแบบข้อมูล รวมทั้งมีความสามารถในการดึงข้อมูลจากแหล่งต่างๆ โดยประสานงานกับเว็บเซอร์วิสชนิดต่างๆและแสดงผลออกมาในรูปแบบที่ง่ายต่อการนำไปศึกษาหรือใช้งานต่อได้ และได้รับการตรวจสอบจากนักชีวสารสนเทศแล้วว่าทำงานได้ถูกต้องตรงตามความต้องการและเร็วกว่าการทำงานตามปกติ

แต่จากการทดลองพบว่าบางส่วนของระบบงานในเวิร์คโฟลว์ ยังทำงานได้ไม่ถูกต้องเนื่องจากมีความผิดพลาดมาจากการล้าสมัยของบริการ ทำให้ผู้ใช้งานต้องใช้เวลายาวนานในการตรวจหาสาเหตุของความผิดพลาดที่แท้จริง ดังนั้นหากเวิร์คโฟลว์ได้รับการตรวจสอบความถูกต้องก่อนการทำงานจริงก็จะสามารถลดเวลาที่อาจจะเสียเป็นชั่วโมงๆหรือเป็นวันได้ โดยในวิทยานิพนธ์นี้จะมุ่งเน้นไปที่บริการไบโอมาร์ทในเวิร์คโฟลว์สำหรับวิเคราะห์ของมนุษย์ เพราะเป็นบริการหลักที่ใช้สำหรับคิวรีข้อมูลซึ่งจะกล่าวในบทต่อไป

บทที่ 5

การออกแบบ การพัฒนาและการทดสอบเวิร์คโฟลว์ สำหรับตรวจสอบความถูกต้องของบริการไอโบมาร์ท

5.1 เกริ่นนำ

โปรแกรมทาเวอร์นารุ่นปัจจุบันคือรุ่น 1.7.1 มีปลักอินที่น่าสนใจตัวหนึ่งคือ *Taverna 2 Preview* ซึ่งเป็นการนำคุณสมบัติการทำงานที่เด่นๆของโปรแกรมทาเวอร์นารุ่น 2 ที่กำลังพัฒนาอยู่มาให้ทดลองใช้งานก่อน และนอกจากนี้โครงการมายกริดซึ่งเป็นผู้พัฒนาโปรแกรมทาเวอร์นายังสร้างกลุ่มผู้ทดสอบ *Taverna 2 Preview* เพื่อแลกเปลี่ยนความเห็น และรับฟังข้อคิดเห็นจากผู้ใช้งานอีกด้วย

จากที่เกริ่นนำไว้ในบทก่อนหน้าว่า ปลักอิน *Taverna 2 Preview* มีฟังก์ชันการทำงาน *Health check report* แต่พบว่าไม่สามารถตรวจสอบความผิดพลาดของเวิร์คโฟลว์ที่สร้างขึ้นเพื่อวิเคราะห์สนิบของมนุษย์ได้ เนื่องจากลักษณะของการออกแบบและการทำงานของเวิร์คโฟลว์มีการทำงานที่ซับซ้อน ประกอบด้วยบริการชนิดต่างๆที่หลากหลายและมีพอร์ตอินพุตและเอาท์พุตจำนวนมาก ทั้งยังมีสถาปัตยกรรมการทำงานแบบโมดูลของเวิร์คโฟลว์ที่ซ้อนกันได้หลายโมดูล (Nested workflow) ทำให้ปลักอิน *Taverna 2 Preview* ไม่สามารถตรวจสอบลงไปถึงโมดูลภายในของเวิร์คโฟลว์ได้ ผู้พัฒนาเวิร์คโฟลว์จึงต้องตรวจสอบเวิร์คโฟลว์เองทีละโมดูลๆโดยใช้ปลักอิน *Taverna 2 Preview* แต่ถึงแม้ว่าจะตรวจสอบทีละโมดูลๆ *Taverna 2 Preview* ก็ยังไม่สามารถตรวจสอบค่าการปรับแต่งต่างๆของบริการไบโอมาร์ท ที่เกิดขึ้นเป็นระยะๆเพื่อให้บริการทันสมัยได้ ซึ่งบริการไบโอมาร์ทเป็นบริการหลักที่เลือกมาใช้ในเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

เนื่องจากการบริการไบโอมาร์ทมีการปรับปรุงให้ทันสมัยอยู่เสมอ (Upgrade) ทำให้เวิร์คโฟลว์ที่สร้างไว้ก่อนหน้านั้นทำงานได้ไม่ตรงกับความต้องการ คือ ไม่ผลิตผลลัพธ์ใดๆแต่ก็ไม่ได้แจ้งสถานะการทำงานที่ผิดพลาด จากการตรวจสอบพบว่าแท้จริงแล้วปัญหาเกิดจากการเปลี่ยนแปลงของบริการไบโอมาร์ท ซึ่งเป็นการเปลี่ยนแปลงในระดับฟิลเตอร์และแอ็ตตริบิวต์ ก็ต้องใช้เวลานานในการตรวจสอบ คืออาจจะเป็นวันๆหรือหลายชั่วโมง ยิ่งกว่านั้นผู้วิจัยพบว่า เมื่อรู้

แล้วว่าบริการไบโอมาร์ทใดที่ล้าสมัยแล้ว ไม่ว่าผู้ใช้งานจะพยายามปรับแต่งวิธีการเรียกใช้บริการไอโบมาร์ทอย่างไรก็ตาม ก็ไม่สามารถแก้ปัญหาได้ ทำให้ยิ่งเสียเวลาไปกับการปรับแต่งอีก เพราะยังไม่มีองค์ความรู้หรือคำแนะนำวิธีแก้ไขปัญหาที่ถูกต้อง

เนื้อหาของบทนี้จะกล่าวถึงแนวคิดในการออกแบบ การพัฒนาและการทดสอบเวิร์คโฟลว์ เพื่อตรวจสอบการเปลี่ยนแปลงของบริการไบโอมาร์ทในเวิร์คโฟลว์สำหรับวิเคราะห์สนิปแบบอัตโนมัติ โดยจะสามารถระบุได้ว่าบริการและฟิลเตอร์หรือแอ็ตตริบิวต์ตัวใดที่ล้าสมัยไปแล้วได้ และนั่นย่อมหมายถึงบริการไบโอมาร์ทนั้นได้ล้าสมัยไปแล้วเช่นกัน

5.2 แนวคิดการออกแบบเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท

ในส่วนนี้ จะอธิบายถึงแนวคิดการออกแบบเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ทโดยจะให้รายละเอียดโครงสร้างของ Scufl ซึ่งเป็นเอกสารที่ใช้บันทึกเวิร์คโฟลว์และโครงสร้างของบริการไบโอมาร์ทที่ใช้ในงานวิจัย ดังนี้

5.2.1 โครงสร้างของ Scufl

Scufl ย่อมาจาก Simple conceptual unified flow language เป็นภาษาและมาตรฐานของโปรแกรมทาเวอร์นาที่มายกริดสร้างขึ้นมา [9] เอกสาร Scufl อธิบายอิลิเมนท์และคุณสมบัติที่ปรากฏในไฟล์เวิร์คโฟลว์โดยใช้ไวยากรณ์ XScufl XML ไวยากรณ์นี้ใช้ในโปรแกรม ทาเวอร์นาในการบันทึกหรือเรียกใช้การประกาศโครงสร้างของเวิร์คโฟลว์ โครงการมายกริดไม่สนับสนุนการใช้การภาษานี้นอกเหนือจากสิ่งแวดล้อมของโปรแกรมทาเวอร์นา

- **Scufl top level tag**

  Tag ระดับบนสุดสำหรับการประกาศ Scufl หนึ่งๆ ประกอบด้วยอิลิเมนท์ต่างๆที่เป็นสมาชิกในเวิร์คโฟลว์คือ บริการ (Processor), การเชื่อมโยง (Link), แหล่งผลิตข้อมูล (Source), แหล่งใช้ข้อมูล (Sink) และการประสานงาน (Coordination) สำหรับงานวิจัยนี้จะเป็นการวิเคราะห์อิลิเมนท์ที่อยู่ใน Tag

processor เท่านั้น Tag ต่างๆเหล่านี้จะถูกอธิบายภายใต้เนมสเปซ (Namespace) http://org.embl.ebi.escience/xscufl/0.1alpha โดยมีโครงสร้างดังนี้

- <**scufl** version(*string*) log(*int*)> ประกาศว่าเป็นเอกสาร Scufl และระบุรุ่นของภาษา
  - <processor>* ประกาศกระบวนการทำงานแบบอะตอมมิกในเวิร์ค-โฟลว์
  - <link>* ประกาศการเชื่อมโยงของพอร์ตข้อมูลอินพุตและเอาท์พุตต่างๆในเวิร์คโฟลว์
  - <source>* ประกาศพอร์ตข้อมูลที่ทำหน้าเป็นอินพุต
  - <sink>* ประกาศพอร์ตข้อมูลที่ทำหน้าที่เป็นเอาท์พุต
  - <coordination>* ประกาศการเชื่อมโยงแบบควบคุมของบริการภายในเวิร์คโฟลว์

- **โหนดบริการ**

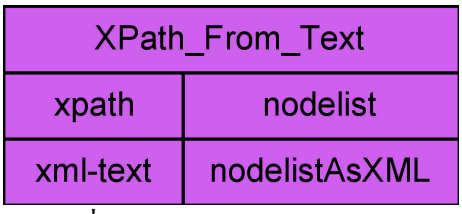
การตรวจสอบบริการไบโอมาร์ทที่เสนอในวิทยานิพนธ์ มีข้อมูลที่เกี่ยวข้องคือ Processor name หรือชื่อบริการ, Dataset name หรือชื่อดาตาเซ็ต, Filter name หรือชื่อฟิลเตอร์และ Attribute name หรือชื่อแอ็ตตริบิวต์ซึ่งอิลิเมนท์ของข้อมูลเหล่านี้ เป็นสมาชิกของโหนดบริการ(Processor node) ทั้งสิ้น โครงสร้างของโหนดบริการคือ

- <processor name(string)> ชื่อบริการ โดยค่าปริยายชื่อบริการจะเป็นชื่อเดียวกับดาตาเซ็ต และผู้ใช้งานสามารถเปลี่ยนชื่อให้สื่อความหมายตามการทำงานหรือตามความต้องการได้
  - <description>?
  - textual description (PCDATA) คำอธิบายการทำงานของบริการ
  - <SPEC ELEMENT maxretries(int)? retrydelay(int)? retrybackoff(double)?> หน่วยการทำงานของบริการ ได้แก่ ค่าคงที่, บริการแบบ WSDL, บริการท้องถิ่น, บริการ Soaplab, บริการ

BioMoby หรือบริการ BioMart และรวมถึง โมดูลของเวิร์คโฟลว์ที่ซ้อนได้หลายโมดูล (Nested workflow) และคุณสมบัติการทำงานซ้ำของบริการเหล่านี้หรือ Retry ค่าโดยปริยายคือ maxretries และ retrydelay = 0 ส่วน retrybackoff = 1

- <iterationstrategy>? กลไกการทำซ้ำของพอร์ตข้อมูลสำหรับการป้อนเป็นอินพุตมีสองชนิดคือ cross product (หลายต่อหลาย) และ Dot product (หนึ่งต่อหนึ่ง) โดยมี Namespace คือ http://org.embl.ebi.escience/ xscufliteration/0.1beta10

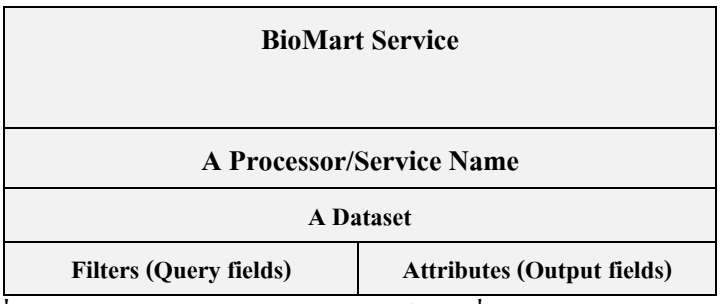
เวิร์คโฟลว์ที่สร้างด้วยโปรแกรมทาเวอร์นานั้น จะบันทึกเป็นไฟล์เอกสาร Scufl ซึ่งเป็นมาตรฐานที่มายกริดสร้างขึ้น อาจกล่าวได้ว่า Scufl เป็นเอกสาร XML ชนิดหนึ่งก็ได้ เนื่องจากมีโครงสร้างแบบภาษา XML จึงสามารถใช้ XML Path Language (XPath) [58] เข้าไปดึงข้อมูลที่ต้องการต่างๆของบริการไบโอมาร์ทที่อยู่ในเอกสาร Scufl ได้ โปรแกรมทาเวอร์นาเองก็มีบริการท้องถิ่นคือ *'XPath From Text'* ไว้ให้ทำงานดังกล่าวได้ดังรูปที่ 5.1 ซึ่งสามารถใส่อินพุตเป็นเอกสาร XML และ XPath expression เพื่อดึง (Extract) ข้อมูลจากเอกสาร XML อินพุตนั้นได้เลย โดยในวิทยานิพนธ์นี้ เอกสาร XML ดังกล่าวก็คือเอกสารเวิร์คโฟลว์หรือเอกสาร Scufl ของเวิร์ค-โฟลว์



รูปที่ 5.1 บริการท้องถิ่น *'XPath From Text'* ในโปรแกรมทาเวอร์นา

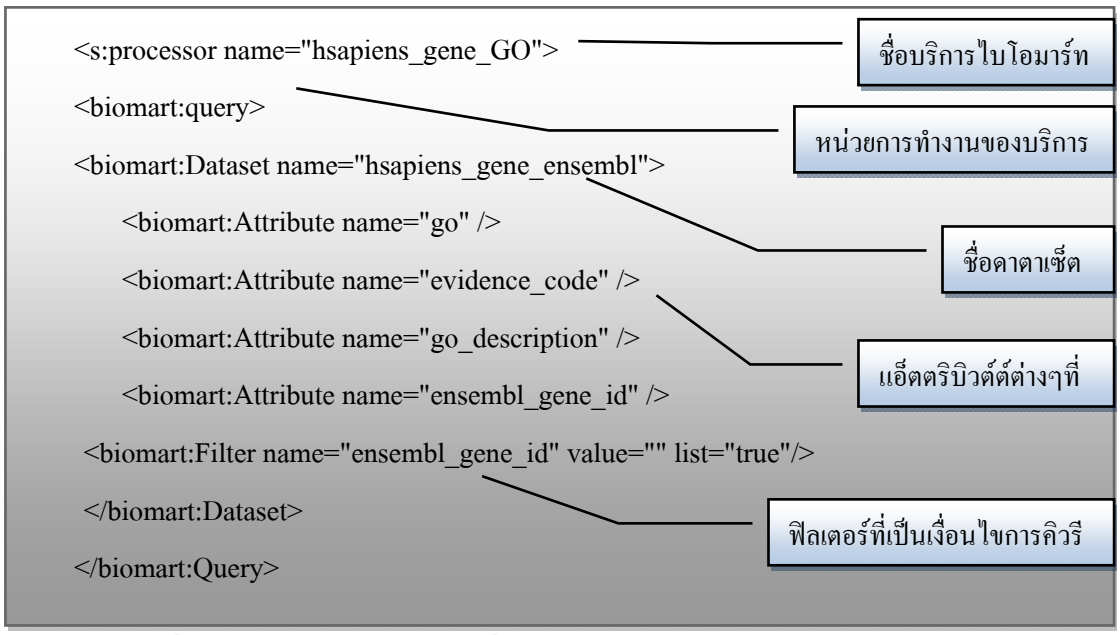
5.2.2 โครงสร้างของบริการไบโอมาร์ทที่ใช้งานวิจัย

บริการไบโอมาร์ทหนึ่งๆจะประกอบด้วยหลายคอมโพเนนท์ หนึ่งในนั้นคือชื่อของบริการหรือ Processor/Service name เนื่องจากบริการไบโอมาร์ททำงานในลักษณะคล้ายเหมืองข้อมูล (Data mining) [12] ดังนั้น บริการไบโอมาร์ทสามารถคิวรีข้อมูล จากฐานข้อมูลหรือดาตาเซ็ตที่ผู้ให้บริการกำหนดไว้ได้ โดยในแต่ดาตาเซ็ตก็จะประกอบด้วยฟิลเตอร์ (ฟิลด์ข้อมูลของเงื่อนไขในการคิวรี) และแอ็ตตริบิวต์ (ฟิลด์ข้อมูลของเอาท์พุตจากการคิวรี) ที่แตกต่างกันออกไป งานวิจัยนี้ได้ตีกรอบเนื้อหาการตรวจสอบของบริการไบโอมาร์ทไว้ว่า บริการหนึ่งๆจะมีหนึ่งชื่อบริการ และสามารถคิวรีข้อมูลได้หนึ่งดาตาเซ็ต เพื่อให้เป็นไปตามแนวคิดการทำงานแบบอะตอมมิกหรือปรมาณูของเว็บเซอร์วิส และจะง่ายต่อการดีบักหรือตรวจสอบการทำงาน และดาตาเซ็ตหนึ่งๆประกอบด้วยหลายฟิลเตอร์และหลายแอ็ตตริบิวต์ ดังแสดงโครงสร้างบริการไบโอมาร์ทดังรูปที่ 5.2



รูปที่ 5.2 โครงสร้างของบริการไบโอมาร์ทในสิ่งแวดล้อมของทาเวอร์นา

ตัวอย่างเอกสาร Scufl ที่มีการบันทึกอิลิเมนท์ของบริการไบโอมาร์ทเช่น ชื่อบริการคือ *hsapiens_gene_GO* โดยคิวรีที่ดาตาเซ็ตชื่อ *hsapiens_gene_ensembl* และมีฟิลเตอร์คือ *ensembl_gene_id* และแอ็ตตริบิวต์ต่างๆคือ *go*, *evidence_code*, *go_description*, และ *ensemble_gene_id* ดังรูปที่ 5.3



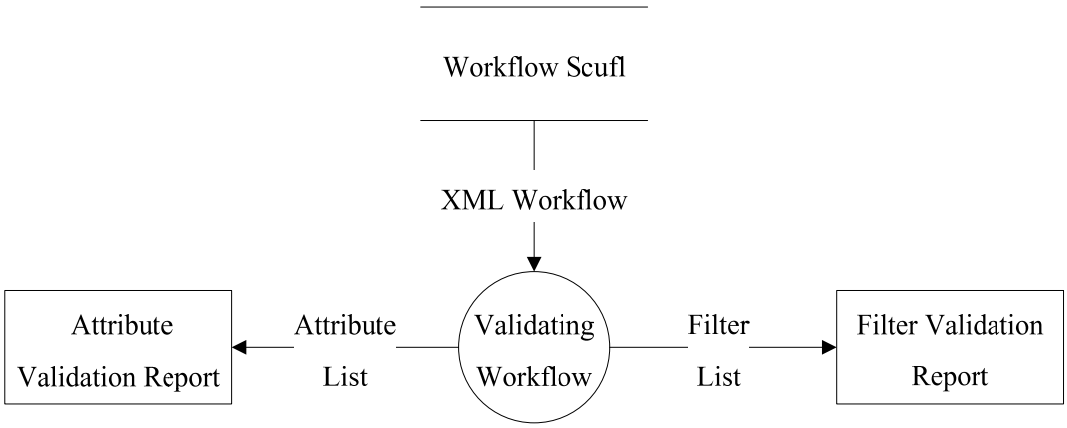
รูปที่ 5.3 ตัวอย่างเอกสาร Scufl ที่มีการบันทึกอิลิเมนท์ของบริการไบโอมาร์ท

อนึ่ง จากกลไกการทำงานของโปรแกรมทาเวอร์นาเอง เมื่อเปิดเวิร์คโฟลว์ใดขึ้นมา โปรแกรมจะทำการตรวจสอบการตอบกลับของเอนพอยต์โดยอัตโนมัติ (Responding) ซึ่งหากเอนพอยต์ของบริการใด ไม่มีการตอบกลับ โปรแกรมทาเวอร์นาจะแจ้งเตือน และไม่สามารถเปิดเวิร์คโฟลว์นั้นขึ้นมาให้ทำงานได้โดยปริยาย [8][9] ดังนั้นในการออกแบบการตรวจสอบบริการไบโอมาร์ท ในวิทยานิพนธ์นี้จึงไม่ได้สนใจการตรวจสอบในเรื่องนี้อีก เพราะถือว่าโปรแกรมทาเวอร์นาได้ทำหน้าที่ตรวจสอบเรื่องนี้อยู่แล้ว แต่จะเน้นความสนใจไปที่ การตรวจสอบฟิลเตอร์และแอ็ตตริ-บิวต์ของดาตาเซ็ตที่เกี่ยวข้องกับบริการไบโอมาร์ทนั้นๆ

5.3 การออกแบบ Data flow ของเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท

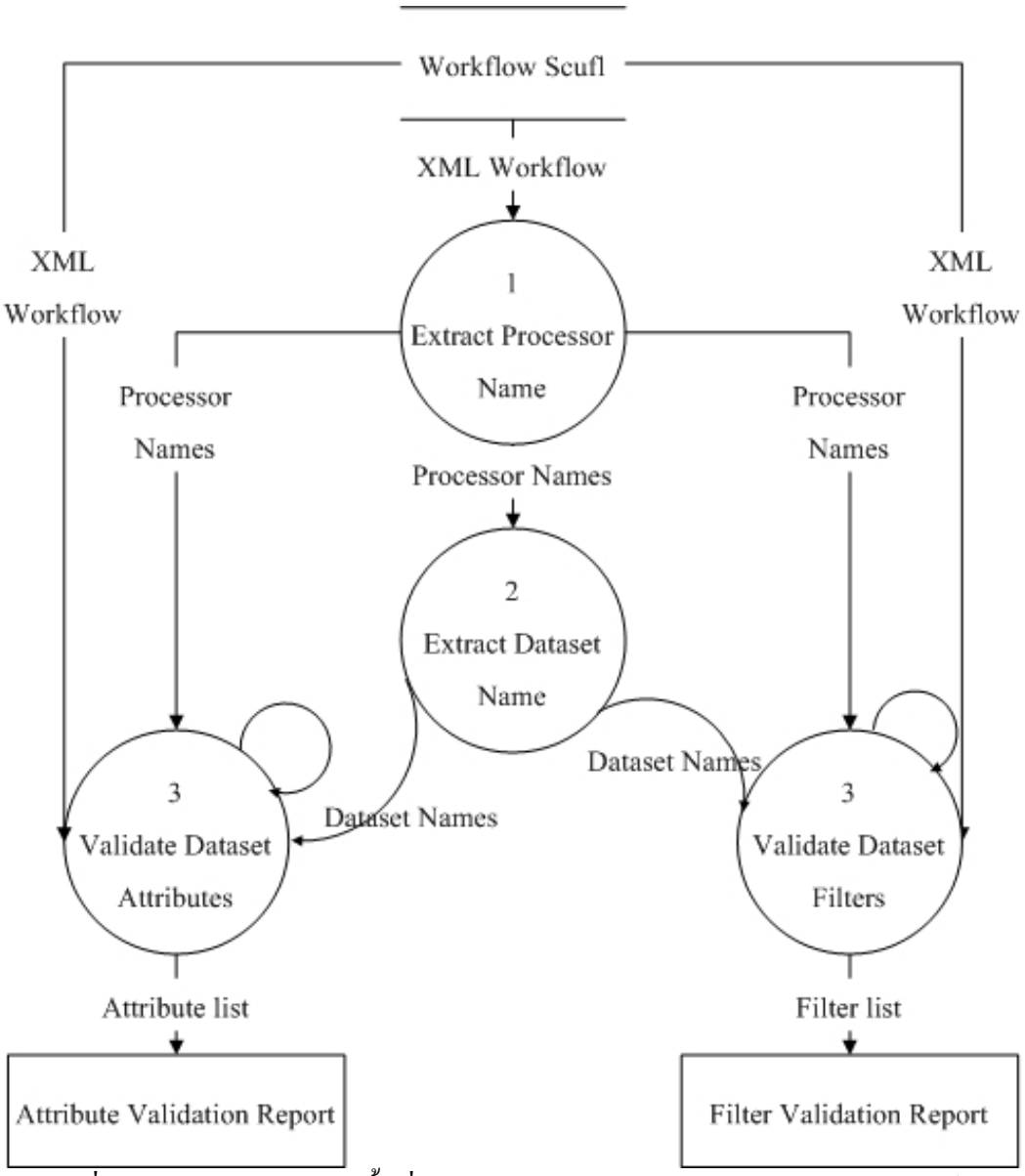
ในขั้นนี้จะอธิบายการออกแบบการทำงานของเวิร์คโฟลว์ ที่จะใช้ในการตรวจสอบบริการไบโอมาร์ทในรูปแบบ Data flow diagram เพื่อให้เห็นการไหลของข้อมูลภาพในเวิร์คโฟลว์ ประกอบด้วย การทำงานของเวิร์คโฟลว์ในภาพรวมหรือ Context diagram และกระบวนการตรวจสอบบริการไบโอมาร์ทดังนี้

5.3.1 การทำงานของเวิร์คโฟลว์ในภาพรวม



รูปที่ 5.4 Context diagram ของระบบการตรวจสอบของเวิร์คโฟลว์

จากรูปที่ 5.4 แสดง Context diagram ของระบบการตรวจสอบเวิร์คโฟลว์ ข้อมูลอินพุตคือเอกสาร Scufl ของเวิร์คโฟลว์ที่มีการเรียกใช้งานบริการไบโอมาร์ท โดยอาจจะอยู่ในรูปแบบของเวิร์คโฟลว์หรือโมดูลของเวิร์คโฟลว์ที่ซ้ำซ้อนกันหลายโมดูล (Nested workflow) ก็ได้ เพื่อนำเข้าสู่กระบวนการตรวจสอบเวิร์คโฟลว์ แล้วจะแจ้งผลลัพธ์เป็นรายงานการตรวจสอบบริการไบโอมาร์ทเพื่อให้ผู้ใช้งานนำข้อมูลไปตัดสินใจเกี่ยวกับการทำงานต่อไปได้ โดยกระบวนการการทำงานของการตรวจสอบจะอธิบายในหัวข้อถัดไป

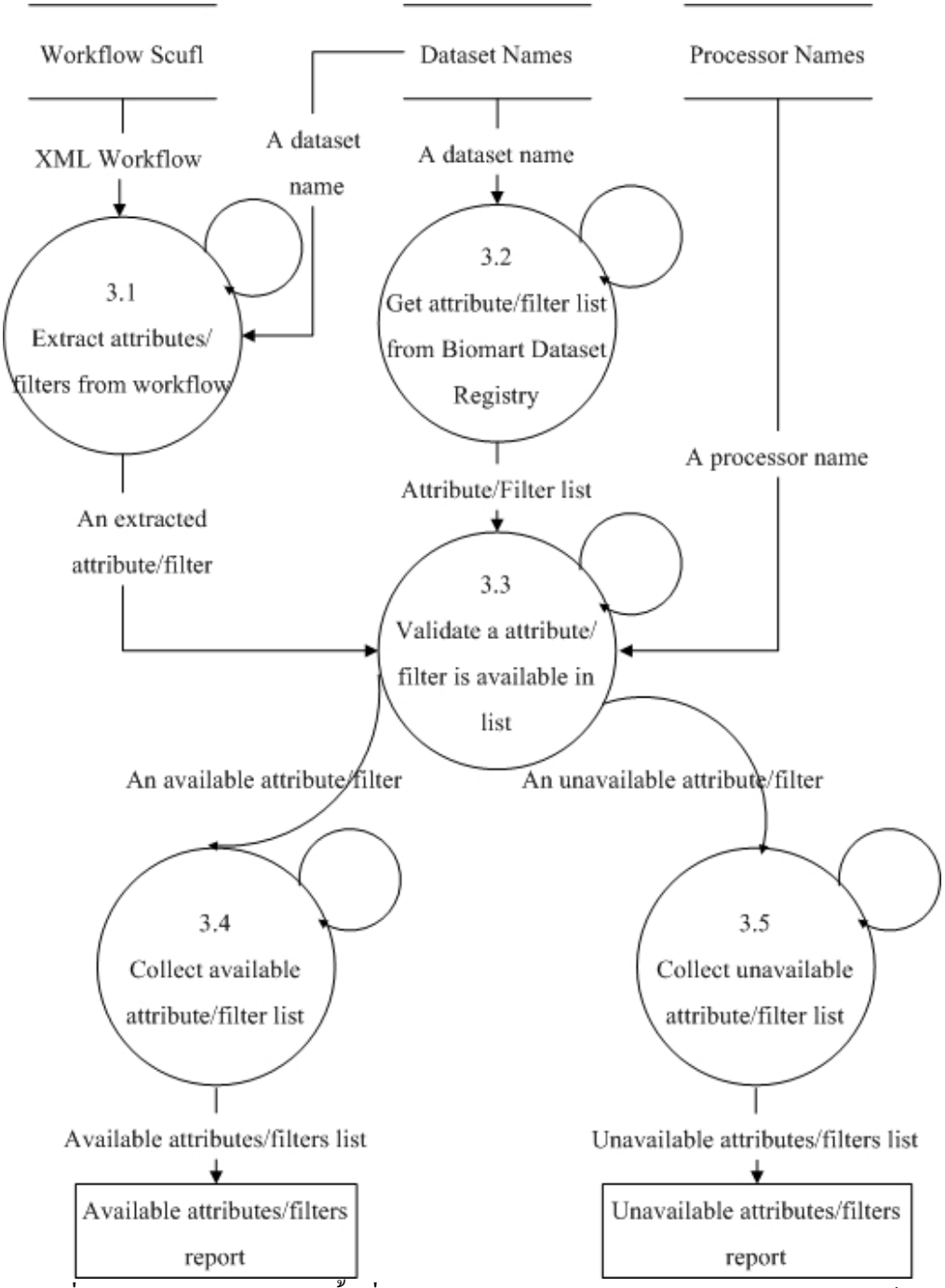


รูปที่ 5.5 Data flow diagram ขั้นที่ 2 อธิบายกระบวนการตรวจสอบบริการไบโอมาร์ท

จากรูปที่ 5.5 แสดงขั้นตอนและการไหลของข้อมูลในการตรวจสอบบริการไบโอ-มาร์ทโดยมีอินพุตคือเอกสาร Scufl ของเวิร์คโฟลว์ จากนั้นทำการดึง (Extract) ชื่อของบริการไบโอ-มาร์ททุกตัวที่อยู่ในเวิร์คโฟลว์ เรียกว่า Processor name แล้วใช้ Processor name ไปหาชื่อดาตาเซ็ต อีกครั้งหนึ่ง เมื่อได้ชื่อดาตาเซ็ตของบริการนั้นๆแล้ว ก็ใช้ชื่อดาตาเซ็ตเป็นคำสำคัญเพื่อเข้าไปตรวจสอบทั้งฟิลเตอร์และแอ็ตตริบิวต์ เพื่อหาว่าข้อมูลใดที่ยังทันสมัยอยู่ และข้อมูลใดที่ล้าสมัยแล้ว

กลไกการตรวจสอบฟิลเตอร์และแอ็ตตริบิวต์ที่มีการทำงานแบบวนซ้ำหรือลูป สามารถอธิบายได้ดังรูปที่ 5.6

5.3.2 กลไกการตรวจสอบบริการไบโอมาร์ท



รูปที่ 5.6 Data flow diagram ขั้นที่ 3 อธิบายกระบวนการตรวจสอบบริการไบโอมาร์ท

จากรูปที่ 5.6 เป็นกลไกในการทำหน้าตรวจสอบบริการไบโอมาร์ท โดยมีอินพุต 3 ชนิดคือ เอกสาร Scufl ของเวิร์คโฟลว์, ชื่อบริการไบโอมาร์ททั้งหมด และชื่อดาตาเซ็ตของบริการนั้นๆ การทำงานจะเริ่มด้วยการส่งข้อมูลเข้าไปทีละบริการหรือทีละดาตาเซ็ต เพราะหนึ่งบริการจะมีเพียงหนึ่งดาตาเซ็ตเท่านั้น ดังนั้นการทำงานจึงมีลักษณะเป็นลูปหรือวนทำงานซ้ำๆจนกว่าข้อมูลอินพุตจะหมด โดยจะใช้ชื่อดาตาเซ็ตเข้าไปดึงอิลิเมนต์ของฟิลเตอร์และแอ็ตตริบิวต์ที่อยู่ในเอกสาร Scufl ของเวิร์คโฟลว์ที่ต้องการจะตรวจสอบ ในอีกด้านหนึ่งก็ใช้ชื่อดาตาเซ็ตอีกเช่นกัน ในการไปสืบค้นข้อมูลรายชื่อฟิลเตอร์และแอ็ตตริบิวต์ ที่ทันสมัยจากฐานข้อมูลรีจิสทรีของไบโอมาร์ทที่ให้บริการบนเครือข่ายอินเตอร์เน็ต

เมื่อได้ข้อมูลทั้ง 2 ส่วนแล้ว จึงเปรียบเทียบว่า ฟิลเตอร์และแอ็ตตริบิวต์ใดๆ จากเอกสาร Scufl ของเวิร์คโฟลว์นั้น ปรากฏอยู่ในรายชื่อฟิลเตอร์และแอ็ตตริบิวต์ของรีจิสทรีไบโอ-มาร์ทหรือไม่ หากปรากฏก็แสดงว่า ฟิลเตอร์และแอ็ตตริบิวต์นั้นๆยังคงทันสมัยอยู่ ก็จะรวบรวมข้อมูลเกี่ยวกับฟิลเตอร์และแอ็ตตริบิวต์นี้ ได้แก่ คำอธิบายและรายละเอียดต่างๆ นำเสนอต่อผู้ใช้งานเพื่อเก็บไว้เป็นข้อมูลอ้างอิงสำหรับการตรวจสอบในอนาคต เพราะเมื่อเวลาผ่านไป บริการนั้นๆก็อาจจะล้าสมัยได้อีกเช่นกัน การนำเสนอข้อมูลคำอธิบายฟิลเตอร์และแอ็ตตริบิวต์นี้ ก็จะช่วยให้สามารถสืบค้นหรือเปรียบเทียบได้ในอนาคต

ส่วนฟิลเตอร์และแอ็ตตริบิวต์ใด ที่ไม่ปรากฏในอยู่รายชื่อฟิลเตอร์และแอ็ตตริบิวต์ของรีจิสทรีไบโอมาร์ท แสดงว่า บริการไบโอมาร์ทที่มีฟิลเตอร์และแอ็ตตริบิวต์เหล่านั้นได้ล้าสมัยไปแล้ว จึงไม่มีข้อมูลใดๆที่เกี่ยวข้องกับฟิลเตอร์และแอ็ตตริบิวต์นั้นๆอีก จึงรายงานเพื่อแจ้งให้ผู้ใช้งานทราบ

5.4 การพัฒนาเวิร์คโฟลว์

ในขั้นนี้จะอธิบายขั้นตอนการพัฒนาเวิร์คโฟลว์ (Workflow implementation) โดยจะให้รายละเอียดการดึงอิลิเมนท์ข้อมูลต่างๆในเอกสาร Scufl ของเวิร์คโฟลว์ การสืบค้นข้อมูลรายชื่อฟิลเตอร์และแอ็ตตริบิวต์ที่ทันสมัย การเปรียบเทียบข้อมูลในเวิร์คโฟลว์กับรีจิสทรีของไบโอมาร์ท และการปรับแต่งกลไกการซ้ำของพอร์ตอินพุตในเวิร์คโฟลว์ดังต่อไปนี้

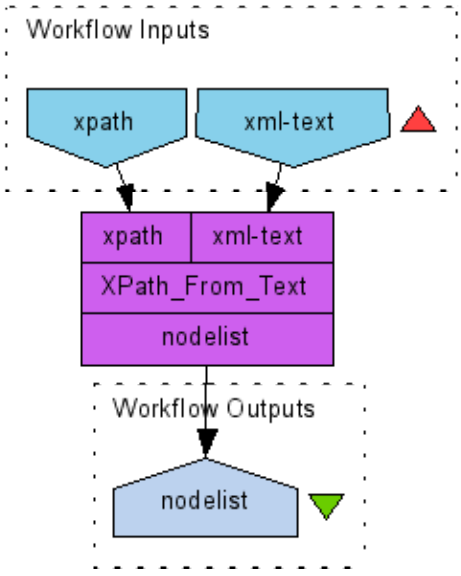
5.4.1 การดึงอิลิเมนท์ข้อมูลในเอกสาร Scufl ของเวิร์คโฟลว์

การดึงอิลิเมนท์ของข้อมูลต่างๆในเอกสาร Scufl ของเวิร์คโฟลว์สามารถใช้ชุดคำสั่งของ XPath เข้าไปดึงอิลิเมนท์ของข้อมูลที่ต้องการได้ ตารางที่ 5.1 แสดงชุดคำสั่ง XPath ในการดึงอีลิเมนท์ข้อมูลที่จะนำมาใช้ในการทำงานตรวจสอบเวิร์คโฟลว์ และโปรแกรมทาเวอร์นาก็มีบริการท้องถิ่นชื่อ *'XPath From Text'* (รูปที่ 5.1) รองรับการทำงานนี้ไว้เรียบร้อยแล้ว

ตารางที่ 5.1 XPaht expression ที่ใช้ดึงอีลิเมนท์ของข้อมูลในเวิร์คโฟลว์

Extracting	Xpath expression
Processor names	//*[(local-name()="Dataset" and (namespace-uri()="http://org.embl.ebi.escience/xScufl-Biomart/0.1alpha")]/ancestor::s:processor/@name
BioMart processor names	//*[(local-name(.)='processor') and (@name=<Processor Name>)] /*[local-name(.)='BioMart']/parent::s:processor/@name
BioMart dataset names	/*[(local-name(.)='processor') and (@name=<A BioMart processor name>)] /*[local-name(.)='BioMart'] /*[local-name(.)='MartQuery'] /*[local-name(.)='Query'] /*[local-name(.)='Dataset']/@name
BioMart dataset filters or attributes	/*[(local-name(.)='processor') and (@name=<A BioMart processor name>)] /*[local-name(.)='BioMart'] /*[local-name(.)='MartQuery'] /*[local-name(.)='Query'] /*[local-name(.)='Dataset'] /*[local-name(.)='Filter' | 'Attribute']/@name

รูปที่ 5.7 แสดงการใช้บริการท้องถิ่นอย่างง่าย *'XPath_From_Text'* ซึ่งมีอินพุต 2 อินพุตได้แก่ XPath expression (xpath) ดังแสดงในตารางที่ 5.1 และเอกสาร Scufl ของเวิร์คโฟลว์ (xml-text) โดยจะได้เอาท์พุตเป็นรายการของอิลิเมนท์ (nodelist) ที่ระบุใน XPath expression นั้นๆ



รูปที่ 5.7 การใช้งานบริการ XPath_From_Text

5.4.2 การสืบค้นข้อมูลรายชื่อฟิลเตอร์และแอ็ตตริบิวต์ที่ทันสมัย

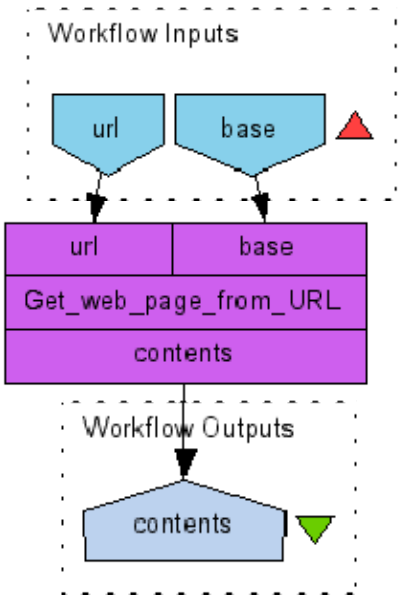
การสืบค้นข้อมูลรายชื่อฟิลเตอร์และแอ็ตตริบิวต์ที่ทันสมัย สามารถทำได้โดยการค้นหาฐานข้อมูลรีจิสทรีกลางของบริการไบโอมาร์ท ชุดเมตาดาตาของดาตาเซ็ตที่ให้บริการสามารถเรียกได้จากการเติมพารามิเตอร์ต่อท้าย URL หลัก (Base URL) พารามิเตอร์แรกคือเครื่องหมาย *'?'* จากนั้นพารามิเตอร์อื่นๆคือเครื่องหมาย *'&'* เมตาดาตาของดาตาเซ็ตเข้าถึงได้โดยการร้องขอด้วยคำสั่ง *GET* ของ HTTP ดังตารางที่ 5.2

ตารางที่ 5.2 รูปแบบ URL สำหรับการสืบค้นรีจิสทรีของบริการไบโอมาร์ท

ประเภท (Type)	**การอิมพลีเมนท์ (Implementation)**
Base URL	http://www.biomart.org
ฟิลเตอร์รีจิสทรี (Filter registry)	/biomart/martservice?type=filters&virtualschema=default&dataset =<datasetname>
แอ็ตตริบิวต์รีจิสทรี (Attribute registry)	/biomart/martservice?type=attributes&virtualschema=default&dataset =<datasetname>

ตัวอย่างการค้นหารายชื่อแอ็ตตริบิวต์ต่างๆของดาตาเซ็ต เช่น ดาตาเซ็ตชื่อ *hsapiens_gene_ensembl* จะมี Base URL คือ http://www.biomart.org และมีแอ็ตตริบิวต์รีจิสทรีคือ /biomart/martservice ?type=attributes&virtualschema=default&dataset=hsapiens_gene_ensembl

โปรแกรมทาเวอร์นามีบริการท้องถิ่นชื่อ *Get_web_page_from_URL* รองรับการทำงานในการร้องขอ *GET* ข้อมูลผ่านเครือข่ายอินเตอร์เน็ตของ HTTP รูปที่ 5.8 แสดงการใช้งานอย่างง่ายของบริการ โดยมีการรับอินพุตสอง 2 อย่างได้แก่ URL หลัก (base) และพารามิเตอร์ (url) จะได้เอาท์พุตคือรายชื่อของฟิลเตอร์และแอ็ตตริบิวต์ต่างๆตามที่ระบุในพารามิเตอร์



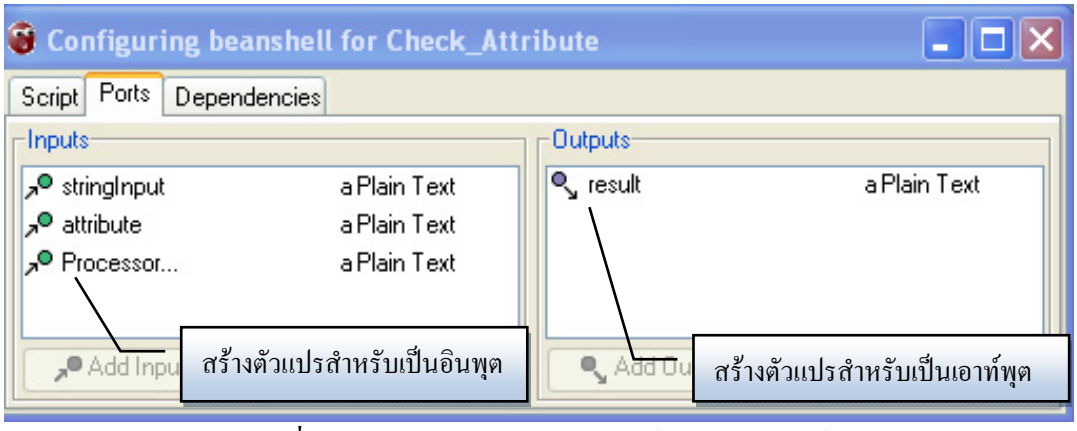
รูปที่ 5.8 ตัวอย่างการใช้งานบริการ *Get_web_page_from_URL*

5.4.3 การเปรียบเทียบข้อมูลในเวิร์คโฟลว์กับรีจิสทรีของไบโอมาร์ท

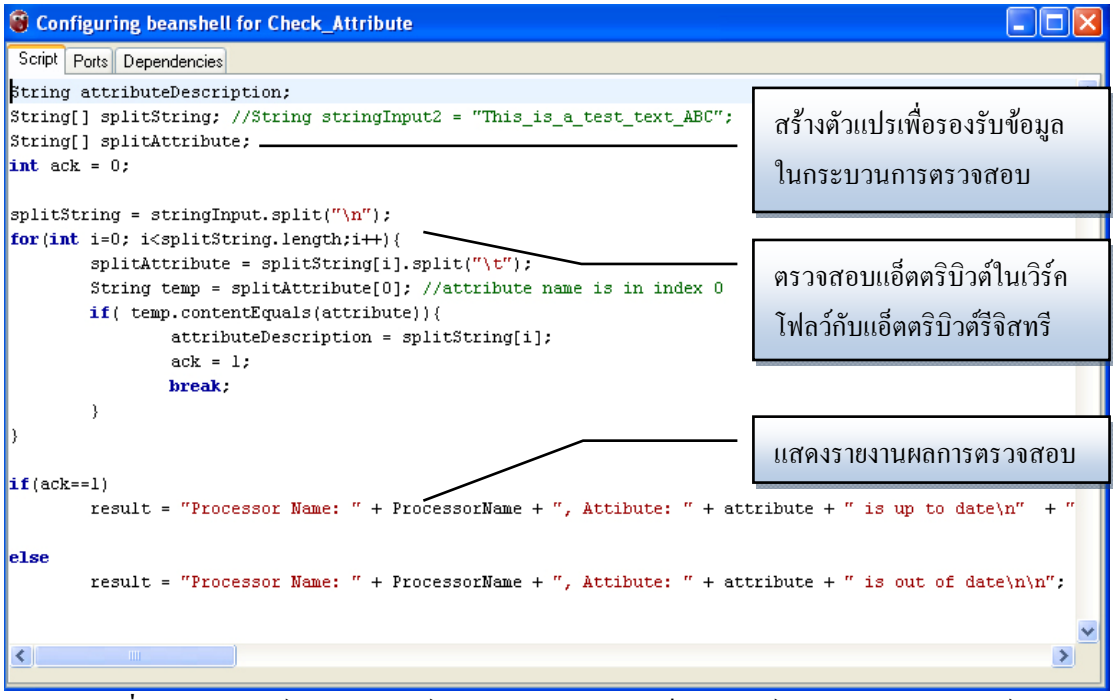
เมื่อสามารถดึงอิลิเมนท์ของข้อมูลต่างๆที่ต้องการจากเอกสาร Scufl ของเวิร์คโฟลว์ และสามารถสืบค้นข้อมูลฟิลเตอร์และแอ็ตตริบิวต์ต่างๆที่ทันสมัยจากฐานข้อมูลรีจิสทรีกลางของบริการไบโอมาร์ทได้แล้ว จึงนำข้อมูลสองส่วนนี้มาเปรียบเทียบกันเพื่อหาว่าฟิลเตอร์และแอ็ตตริบิวต์ใดที่ได้ล้าสมัยไปแล้วหรือยังคงทันสมัยอยู่ กระบวนการทำงานในส่วนนี้ต้องใช้ภาษาคอมพิวเตอร์ในการพัฒนาโปรแกรมเข้ามาช่วยเหลือ โดยภายใต้การพัฒนาเวิร์คโฟลว์ในสิ่งแวดล้อมของโปรแกรมทาเวอร์นา มีบริการท้องถิ่นชื่อว่าบีนเชลล์ซึ่งเป็นลักษณะเชลล์ทำหน้าที่

ตีความสคริปต์ของภาษา Java ได้ ดังนั้นจึงสามารถเขียนอัลกอริทึมในรูปแบบของสคริปต์ฝังไว้ในบีนเชลล์ได้

กระบวนการที่ 3.3 จากรูปที่ 5.6 แสดงให้เห็นว่า กระบวนการรับข้อมูล 3 ชนิดคือ ฟิลเตอร์และแอ็ตตริบิวต์ที่ถูกดึงมาจาก Scufl ของเวิร์คโฟลว์, ข้อมูลรีจิสทรีของดาตาเซ็ตที่ฟิลเตอร์และแอ็ตตริบิวต์เป็นสมาชิก และชื่อของบริการที่เรียกใช้ดาตาเซ็ตนั้นๆ ทำให้สามารถปรับแต่งเพื่อให้บีนเชลล์รับข้อมูลอินพุตได้ 3 พอร์ตและมี 1 พอร์ตสำหรับรายงานผลดังรูปที่ 5.9



รูปที่ 5.9 การกำหนดอินพุตและเอาท์พุตในบีนเชลล์



รูปที่ 5.10 สคริปต์ของบีนเชลล์ในการเปรียบเทียบแอ็ตตริบิวต์ของบริการไบโอมาร์ท

การกำหนดอินพุตและเอาท์พุตในบีนเชลล์ก็คือ แนวคิดการกำหนดตัวแปรเพื่อรองรับข้อมูลที่ไหลเข้าและไหลออกจากบีนเชลล์ เมื่อกำหนดตัวแปรเรียบร้อยแล้วสามารถเขียนสคริปต์ตามอัลกอริทึมในการเปรียบเทียบฟิลเตอร์และแอ็ตตริบิวต์ที่ออกแบบจาก Data flow diagram ได้ ตัวอย่างสคริปต์ของบีนเชลล์ในการตรวจสอบแอ็ตตริบิวต์แสดงดังรูปที่ 5.10

5.4.4 การปรับแต่งกลไกการทำซ้ำของพอร์ตอินพุตในเวิร์คโฟลว์

วัตถุประสงค์ของการทำซ้ำของพอร์ตอินพุตในโปรแกรมทาเวอร์นาก็คือ การจัดการกับกลไกการป้อนข้อมูลอินพุตให้กับกระบวนการการทำงาน ในงานด้านการเขียนโปรแกรมภาษาคอมพิวเตอร์ การจัดการข้อมูลอินพุตเหล่านี้ อาจจะอยู่ในรูปแบบ List หรือรายการข้อมูลที่ประกอบด้วย Item ต่างๆของข้อมูล จากนั้นใช้วิธีการโปรแกรมแบบลูปในการส่งข้อมูลทีละ Item เข้าไปประมวลในกระบวนการทำงานที่ต้องการนั้น

จากการออกแบบใน Data flow diagram ในรูปที่ 5.5 และ รูปที่ 5.6 นั้น แสดงให้เห็นว่า แต่ละกระบวนการมีการทำงานแบบวนซ้ำหรือลูป ดังนั้นจึงมีกลไกการป้อนข้อมูลอินพุตให้กับกระบวนการนั้นๆ โดยมีอินพุตคือ เอกสาร Scufl ของเวิร์คโฟลว์ (Workflw Scufl), ชื่อบริการไบโอมาร์ททั้งหมด (Processor name) และชื่อดาตาเซ็ตของบริการนั้นๆ (Dataset name) การทำงานคือจะส่งข้อมูลเข้าไปทีละบริการหรือทีละดาตาเซ็ต เพราะหนึ่งบริการจะมีเพียงหนึ่งดาตา-เซ็ตเท่านั้น

การจัดการอินพุตของโปรแกรมทาเวอร์นานั้นมี 2 ลักษณะคือ Dot product หรือหนึ่งพอร์ตต่อหนึ่งพอร์ต (One against one) และ Cross product หรือแบบหลายพอร์ตต่อหลายพอร์ต (All against all) ซึ่งค่าโดยปริยายของโปรแกรมทาเวอร์นาเป็นแบบ Cross product นั้นไม่สอดคล้องกับลักษณะโครงสร้าง Scufl ของบริการไบโอมาร์ทที่จะตรวจสอบในวิทยานิพนธ์นี้

ตารางที่ 5.3 แสดงตัวอย่างของข้อมูล 2 Lists แต่ละ List ประกอบด้วย 3 Items หากจัดการข้อมูลให้มีการทำซ้ำกันแบบ Cross product จะได้ผลลัพธ์ดังตารางที่ 5.4 และหากจัดการข้อมูลให้มีการทำซ้ำกันแบบ Dot product จะได้ผลลัพธ์ดังตารางที่ 5.5

ดังนั้นในวิทยานิพนธ์นี้ บริการไบโอมาร์ททั้งหมด (Processor name) และชื่อดาตาเซ็ตของบริการนั้นๆ (Dataset name) จะต้องกำหนดการทำซ้ำกันแบบ Dot product เพราะเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในขณะที่เอกสาร Scufl ของเวิร์คโฟลว์ (Workflow Scufl) จะต้องกำหนดการทำซ้ำกันแบบ Cross product เพราะบริการไบโอมาร์ทและดาตาเซ็ตเป็นสมาชิกในเอกสาร Scufl ดังนั้นจึงสามารถปรับแต่งกลไกการทำซ้ำได้ดังรูปที่ 5.11

ตารางที่ 5.3 ตัวอย่างของข้อมูลอินพุตแบบหลายรายการ

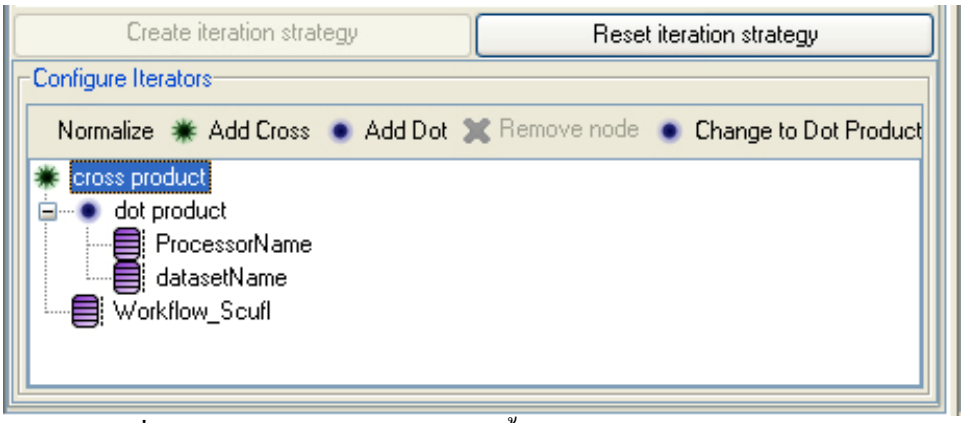
String1	String2
A	B
C	D
E	F

ตารางที่ 5.4 ผลลัพธ์การทำซ้ำแบบ Cross product

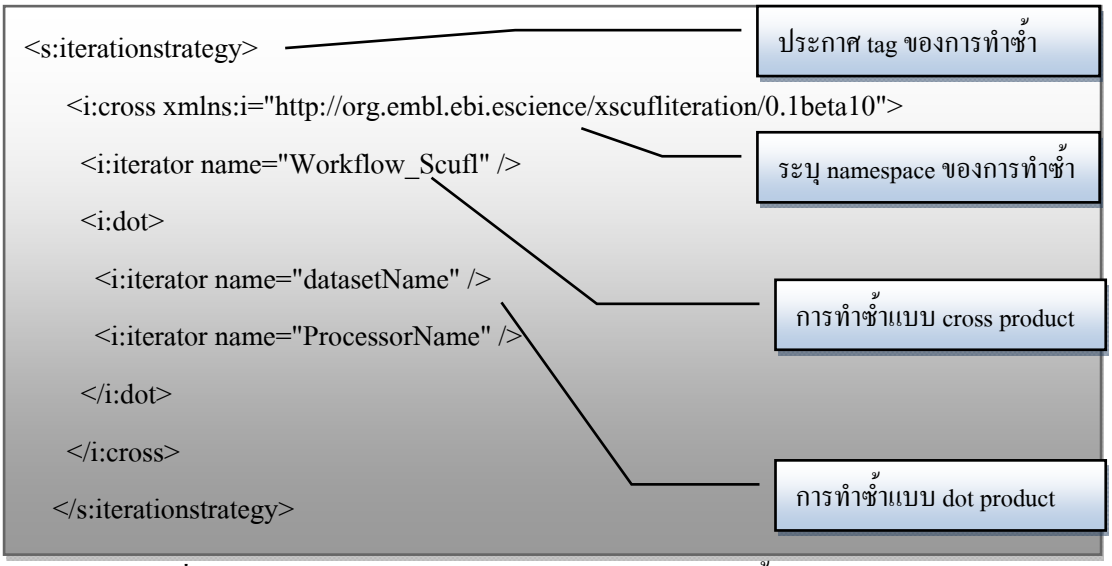
Cross product output
AB
AD
AF
CB
CD
CF
EB
ED
EF

ตารางที่ 5.5 ผลลัพธ์การทำซ้ำแบบ Dot product

Dot product output
AB
CD
EF



รูปที่ 5.11 การปรับแต่งกลไกการทำซ้ำของพอร์ตอินพุตในเวิร์คโฟลว์



รูปที่ 5.12 โครงสร้างของ Scufl ในการปรับแต่งการทำซ้ำของพอร์ตอินพุต

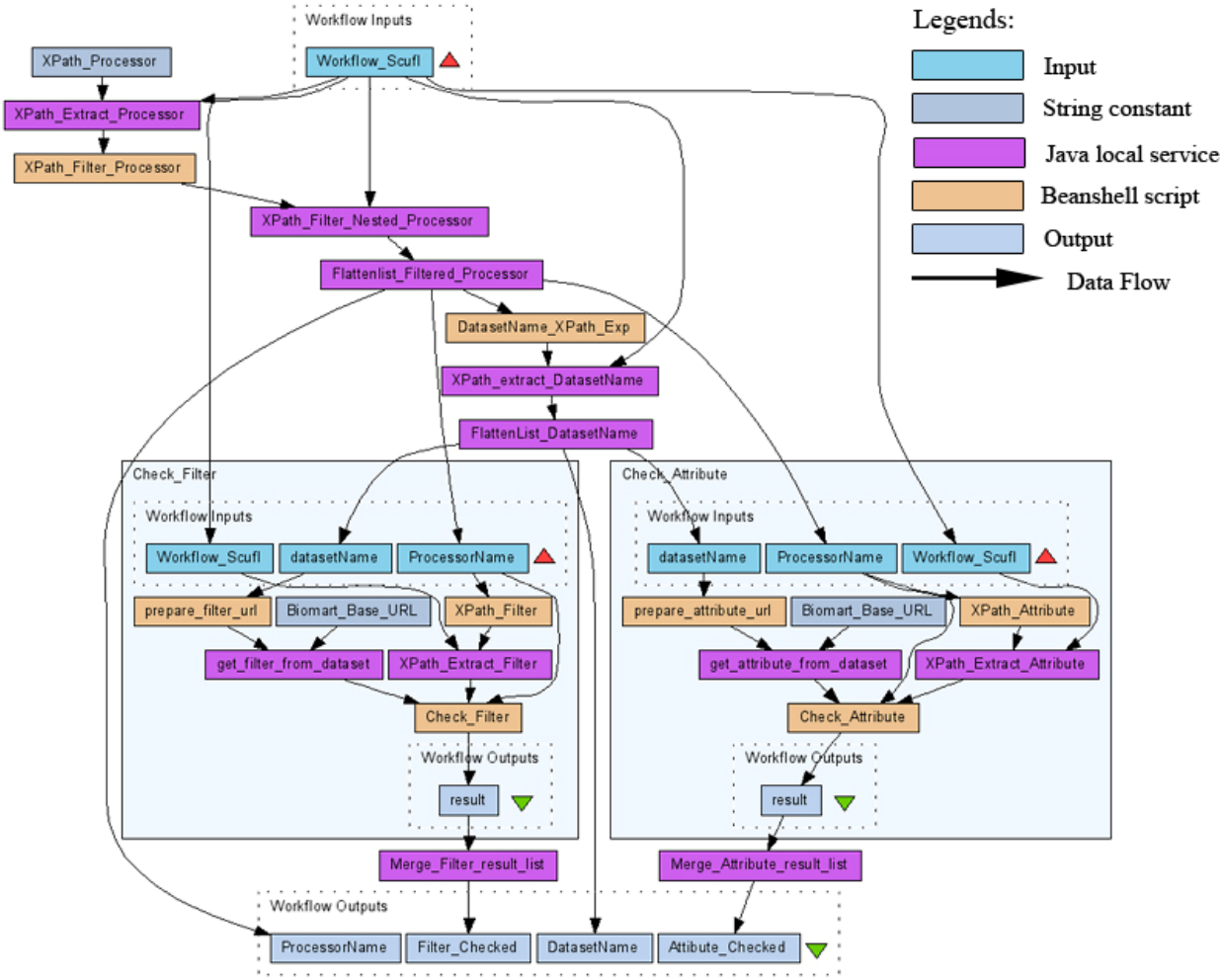
จากการปรับแต่งในรูปที่ 5.11 แสดงความสัมพันธ์ของแต่อินพุตในเวิร์คโฟลว์ เอกสาร Scufl ของเวิร์คโฟลว์ (Workflow Scufl) มีชนิดการทำซ้ำของข้อมูลแบบ Cross product

ส่วนชื่อบริการไบโอมาร์ททั้งหมด (Processor name) และชื่อดาตาเซ็ตของบริการนั้นๆ (Dataset name) มีชนิดการซ้ำกันของข้อมูลแบบ Dot product ผลที่ได้คือรูปที่ 5.12 แสดงโครงสร้างของ Scufl ในการปรับแต่งการทำซ้ำของพอร์ตอินพุตเหล่านี้ในเวิร์คโฟลว์ ดังนั้นผลลัพธ์ความสัมพันธ์ของกลไกการทำซ้ำโดยใช้ข้อมูลอินพุตทั้ง 3 ดังแสดงในตารางที่ 5.6 สามารถอธิบายได้ว่า เอกสาร Scufl ของเวิร์คโฟลว์ใดๆจะประกอบด้วยจำนวนบริการได้หลายบริการ และแต่ละบริการประกอบด้วยดาตาเซ็ตหนึ่งดาตาเซ็ต

ตารางที่ 5.6 ผลลัพธ์กลไกการทำซ้ำของข้อมูลอินพุตในเวิร์คโฟลว์

Cross product	**Dot product**	**Dot product**
Workflow Scufl	1st Dataset name	1st Processor name
Workflow Scufl	2nd Dataset name	2nd Processor name

จากการออกแบบและการพัฒนาตามลำดับขั้นตอนดังกล่าวแล้ว สามารถสร้างเวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ทได้ดังรูปที่ 5.13 และกรณีศึกษาของเวิร์คโฟลว์นี้ก็คือ เวิร์คโฟลว์สำหรับการวิเคราะห์สนิปของมนุษย์ที่กล่าวถึงการออกแบบและการพัฒนาแล้วในบทที่ 4

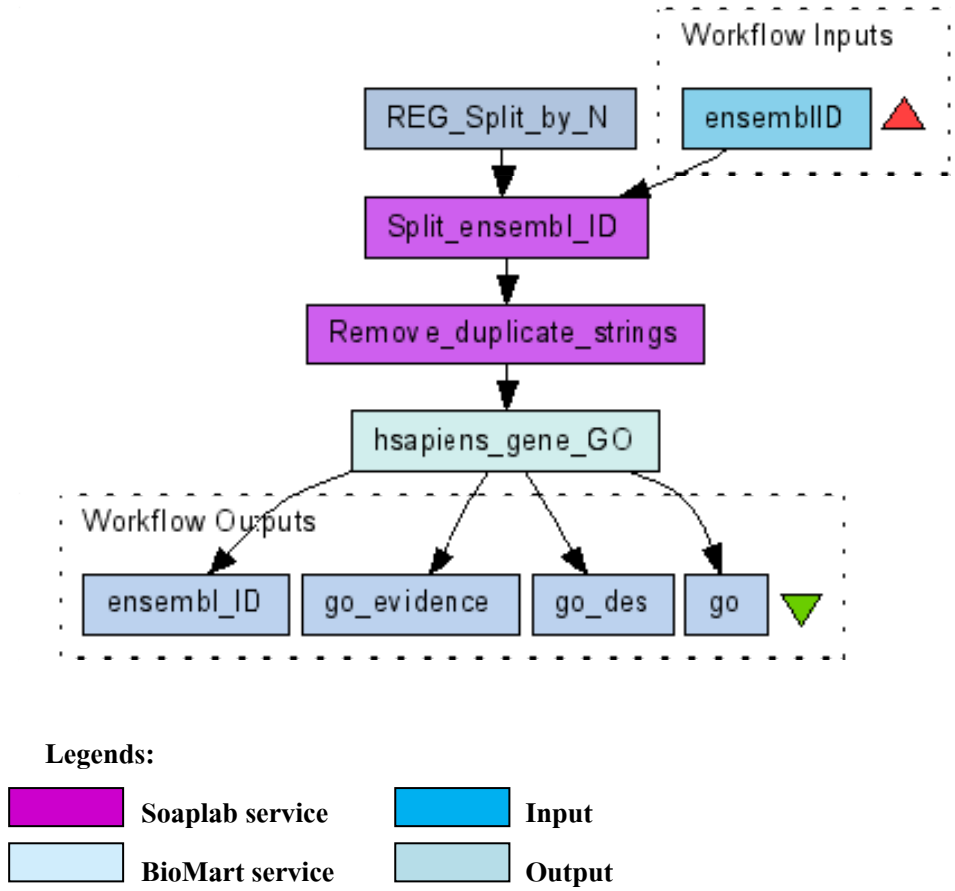


รูปที่ 5.13 เวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ท

5.5 การตรวจสอบเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท

ในเบื้องต้นได้สร้างเวิร์คโฟลว์อย่างง่าย ประกอบด้วยบริการไบโอมาร์ทที่ทำหน้าที่ในการคิวรีข้อมูลจำนวน 1 บริการชื่อ *hsapiens_gene_ensembl* แต่เปลี่ยนชื่อเป็น *hsapiens_gene_GO* เพื่อบอกจุดประสงค์การทำงานค้นหา Gene ontology เวิร์คโฟลว์ประกอบด้วย 1 ฟิลเตอร์สำหรับเป็นเงื่อนไขการคิวรีข้อมูลคือ Ensembl ID และ 3 แอ็ตตริบิวต์สำหรับเป็นข้อมูลเอาท์พุตเกี่ยวกับ Gene Ontology คือ Gene ontology, Gene ontology description และ Gene ontology evidence และอีก 1 แอ็ตตริบิวต์คือ Ensembl ID เอง กล่าวคือ Ensembl ID ที่เป็นอินพุตของฟิลเตอร์คิวรีข้อมูลในดาตาเซ็ต *hsapiens_gene_ensembl* โดยจะคิวรี Ensembl ID มาด้วยเช่นกัน

เพื่อใช้ตรวจสอบความถูกต้องของบริการ เวิร์คโฟลว์ทดสอบอย่างง่ายดังรูปที่ 5.14 และตัวอย่างการทำงานของบริการท้องถิ่น Java ดังตารางที่ 5.7 โดยมีโครงสร้างภาษา Scufl ของเวิร์คโฟลว์ดังรูปที่ 5.15และมีผลการตรวจสอบด้วยเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ทดังนี้



รูปที่ 5.14 เวิร์คโฟลว์อย่างง่ายสำหรับการทดลอง

ตารางที่ 5.7 ตัวอย่างการทำงานของบริการท้องถิ่น Java

ชื่อบริการ	หน้าที่	ตัวอย่างอินพุต	เอาต์พุต
Spilt_ensembl_ID	แยก string	“ENSG00000117242 ENSG00000158828 ENSG00000158828”	“ENSG00000117242” “ENSG00000158828” “ENSG00000158828”
Remove_duplicate _strings	ลบ string ที่ ซ้ำซ้อน	“ENSG00000117242” “ENSG00000158828” “ENSG00000158828”	“ENSG00000117242” “ENSG00000158828”

จากรายงานในรูปที่ 5.16 ทำให้รู้ว่า แอ็ตตริบิวต์ทุกตัวของการทำงานในส่วน Gene ontology นั้นได้ล้าสมัยไปแล้ว (Out-of-date) ซึ่งสามารถตรวจสอบโดยวิธีการแบบตรงไปตรงมาว่าแอ็ตตริบิวต์เหล่านี้ได้ล้าสมัยจริงหรือไม่ โดยตรวจสอบที่หน้าต่างการปรับแต่งของบริการไบโอมาร์ท (Configuration screen) แล้วค้นหาที่หน้าจอนั้นว่า แอ็ตตริบิวต์ดังกล่าวปรากฏบนหน้าต่างหรือไม่ ซึ่งแอ็ตตริบิวต์ที่ล้าสมัยไปแล้ว จะไม่ปรากฏที่หน้าจอการปรับแต่งของบริการไบโอมาร์ทอีก เนื่องจากการเรียกหน้าต่างการปรับแต่งของบริการไบโอมาร์ท ณ ขณะเวลาใดๆที่กำลังใช้งานโปรแกรมทาเวอร์นา โปรแกรมจะใช้ข้อมูลของบริการที่ทันสมัยที่สุด ณ เวลานั้นๆ

นอกจากนี้เวิร์คโฟลว์ที่พัฒนา ยังสามารถให้ข้อมูลที่จำเป็นของฟิลเตอร์และแอ็ตตริบิวต์ที่ทันสมัยสำหรับการตรวจสอบในอนาคตได้อีกด้วย โดยให้เอาท์พุตเป็นรายงานดังรูปที่ 5.17

จากรูปที่ 5.17 รายงานข้างต้นทำให้ทราบว่า ฟิลเตอร์ชื่อ *ensembl_gene_id* มีคำอธิบายว่าเป็นเงื่อนไขการค้นหาข้อมูลโดยสามารถรองรับเป็นรายชื่อของ Ensembl Gene ID ได้ ส่วนแอ็ตตริบิวต์ชื่อ *ensembl_gene_id* ก็คือ Ensembl Stable ID ของยีนนั้นๆซึ่งจะปรากฏอยู่ในหน้า *Feature* ของหน้าจอการปรับแต่งของบริการไบโอมาร์ท ซึ่งสามารถตรวจสอบความถูกต้องของรายงานการทำงานของเวิร์คโฟลว์โดยวิธีการแบบ Manual โดยตรงได้

จากการตรวจสอบโดยวิธีการแบบ Manual แอ็ตตริบิวต์ที่ล้าสมัยจะไม่ปรากฏอยู่ในหน้าจอการปรับแต่งแล้วอันเป็นผลมาจากการปรับปรุงบริการไบโอมาร์ท จึงสามารถสรุปได้ว่าบริการไบโอมาร์ทที่ทดสอบนี้ ได้ล้าสมัยไปแล้ว

```
<?xml version="1.0" encoding="UTF-8"?>
<s:scufl xmlns:s="http://org.embl.ebi.escience/xscufl/0.1alpha" version="0.2" log="0">
 <s:workflowdescription lsid="urn:lsid:net.sf.taverna:wfDefinition:51ca044a-37bd-4f3e-b627-89199bc6b020" author=""
title="GeneOntologyInformation" />
 <s:processor name="REG_Split_by_N" boring="true">
  <s:stringconstant>\n</s:stringconstant>
 </s:processor>
 <s:processor name="hsapiens_gene_GO">
  <s:description>Homo sapiens genes (NCBI36)</s:description>
  <s:biomart>
   <biomart:MartQuery xmlns:biomart="http://org.embl.ebi.escience/xscufl-biomart/0.1alpha">
    <biomart:MartService location="http://www.biomart.org/biomart/martservice" />
    <biomart:MartDataset displayName="Homo sapiens genes (NCBI36)" name="hsapiens_gene_ensembl"
type="TableSet" initialBatchSize="200" maximumBatchSize="50000" visible="false" interface="default" modified="2007-
08-03 16:38:20">
     <biomart:MartURLLocation database="ensembl_mart_46" default="1" displayName="ENSEMBL 46 GENES
(SANGER)" host="www.biomart.org" includeDatasets="" martUser="" name="ensembl" path="/biomart/martservice"
port="80" serverVirtualSchema="default" virtualSchema="default" visible="1" redirect="0" />
    </biomart:MartDataset>
    <biomart:Query virtualSchemaName="default" count="0" uniqueRows="1" softwareVersion="0.6"
requestId="taverna">
     <biomart:Dataset name="hsapiens_gene_ensembl">
      <biomart:Attribute name="go" />
      <biomart:Attribute name="evidence_code" />
      <biomart:Attribute name="go_description" />
      <biomart:Attribute name="ensembl_gene_id" />
      <biomart:Filter name="ensembl_gene_id" value="" list="true" />
     </biomart:Dataset>
    </biomart:Query>
   </biomart:MartQuery>
  </s:biomart>
 </s:processor>
 <s:processor name="Split_ensembl_ID">
  <s:local>org.embl.ebi.escience.scuflworkers.java.SplitByRegex</s:local>
 </s:processor>
 <s:processor name="Remove_duplicate_strings">
  <s:local>org.embl.ebi.escience.scuflworkers.java.StringStripDuplicates</s:local>
 </s:processor>
 <s:link source="ensemblID" sink="Split_ensembl_ID:string" />
 <s:link source="REG_Split_by_N:value" sink="Split_ensembl_ID:regex" />
 <s:link source="Remove_duplicate_strings:strippedlist"
sink="hsapiens_gene_GO:hsapiens_gene_ensembl.ensembl_gene_id_filter" />
 <s:link source="Split_ensembl_ID:split" sink="Remove_duplicate_strings:stringlist" />
 <s:link source="hsapiens_gene_GO:hsapiens_gene_ensembl.ensembl_gene_id" sink="ensembl_ID" />
 <s:link source="hsapiens_gene_GO:hsapiens_gene_ensembl.evidence_code" sink="go_evidence" />
 <s:link source="hsapiens_gene_GO:hsapiens_gene_ensembl.go" sink="go" />
 <s:link source="hsapiens_gene_GO:hsapiens_gene_ensembl.go_description" sink="go_des" />
 <s:source name="ensemblID">
  <s:metadata>
   <s:description>ENSG00000117242
ENSG00000158828
ENSG00000117242</s:description>
  </s:metadata>
 </s:source>
 <s:sink name="go" />
 <s:sink name="go_des" />
 <s:sink name="go_evidence" />
 <s:sink name="ensembl_ID" />
</s:scufl>
```

ชื่อบริการไบโอมาร์ท

ชื่อดาตาเซ็ต

แอ็ตตริบิวต์ที่ต้องการคิว

ฟิลเตอร์ที่ใช้ในการคิวรี

รูปที่ 5.15 โครงสร้างภาษา Scufl ของเวิร์คโฟลว์อย่างง่ายสำหรับการทดลอง

Processor Name: hsapiens_gene_GO, Attibute: go is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: evidence_code is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: go_description is out-of-date

รูปที่ 5.16 รายงานการตรวจสอบบริการไบโอมาร์ทในส่วนของฟิลด์ข้อมูลที่ล้าสมัย

Processor Name: hsapiens_gene_GO, Filter: ensembl_gene_id is up-to-date
Detail: Filter to include genes with supplied list of Ensembl Gene IDs
Processor Name: hsapiens_gene_GO, Attibute: ensembl_gene_id is up-to-date
Detail: Ensembl Stable ID of the Gene, feature_page

รูปที่ 5.17 รายงานการตรวจสอบบริการไบโอมาร์ทในส่วนของฟิลด์ข้อมูลที่ยังทันสมัย

5.6 สรุป

บทนี้กล่าวถึงแนวคิดในการออกแบบ การพัฒนา และการทดสอบเวิร์คโฟลว์สำหรับตรวจสอบการเปลี่ยนแปลงของบริการไบโอมาร์ท ในเวิร์คโฟลว์สำหรับวิเคราะห์ของมนุษย์แบบอัตโนมัติ และเวิร์คโฟลว์ที่พัฒนาขึ้นใช้บริการท้องถิ่นที่โปรแกรมทาเวอร์นาเตรียมไว้ให้ จากการทดสอบด้วยเวิร์คโฟลว์กรณีศึกษาอย่างง่ายพบว่า เวิร์คโฟลว์สำหรับตรวจสอบการเปลี่ยนแปลงของบริการไบโอมาร์ทซึ่งสามารถระบุได้ว่า บริการและฟิลเตอร์หรือแอ็ตตริบิวต์ตัวใดที่ล้าสมัยไปแล้ว เพราะนั่นหมายความว่าบริการไบโอมาร์ทนั้นๆได้ล้าสมัยไปแล้ว ในบทต่อไปกล่าวถึงการทดสอบและผลการทดสอบของเวิร์คโฟลว์ สำหรับตรวจสอบความถูกต้องของบริการไบโอมาร์ทที่พัฒนาขึ้นในบทที่ 4 เป็นกรณีศึกษา

บทที่ 6

การตรวจสอบบริการไบโอมาร์ทในเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

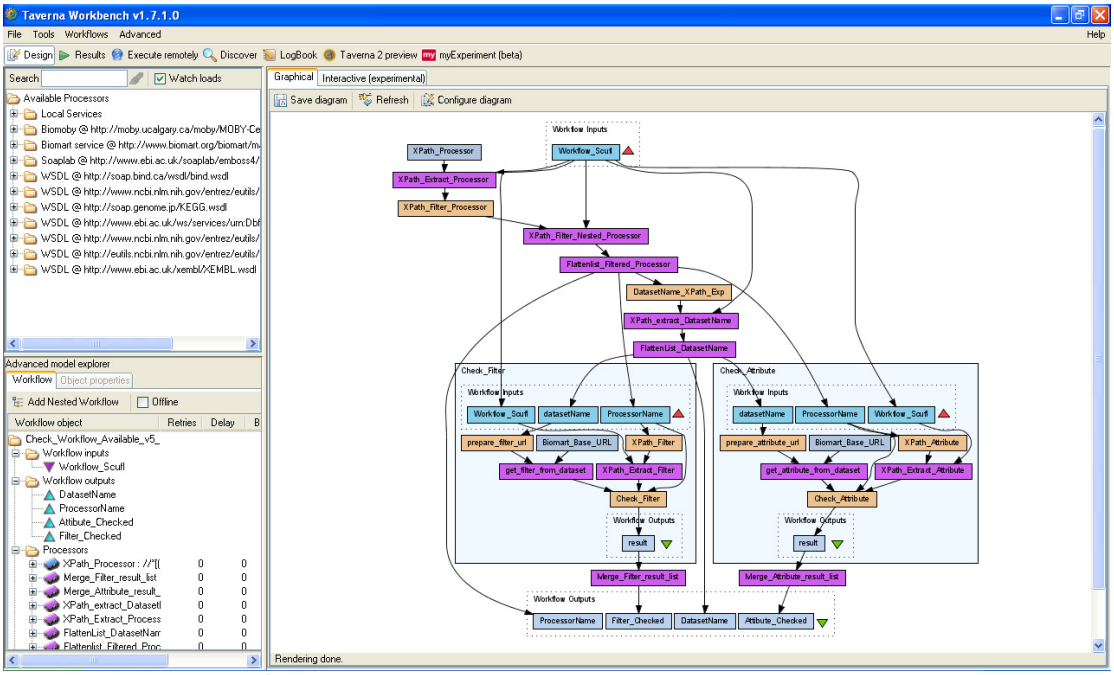
บทนำ

จากบทที่ 4 ได้ทำการทดสอบการทำงานของเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ซึ่งใช้ SNP IDs เป็นอินพุตให้กับเวิร์คโฟลว์จำนวน 1,000 IDs ปรากฏว่าเวิร์คโฟลว์ใช้เวลาทำงานประมาณ 14 ชั่วโมง และผลลัพธ์ที่ได้ก็ไม่ถูกต้องตรงตามความต้องการ ทำให้ต้องเสียเวลาเป็นชั่วโมงๆหรือเป็นวัน ส่วนในบทที่ 5 ได้นำเสนอวิธีการออกแบบและพัฒนาเวิร์คโฟลว์ สำหรับตรวจสอบความถูกต้องของเวิร์คโฟลว์ก่อนการทำงานจริง โดยจะมุ่งเน้นไปที่บริการไบโอมาร์ทเพื่อลดเวลาในการดีบักหรือตรวจสอบเวิร์คโฟลว์ได้อย่างมีนัยสำคัญ ซึ่งผลการทดสอบเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท กับเวิร์คโฟลว์กรณีศึกษาอย่างง่าย พบว่าเวิร์คโฟลว์ที่พัฒนาสามารถระบุบริการไบโอมาร์ทที่ล้าสมัยได้ถูกต้อง ส่วนในบทนี้จะกล่าวถึงการตรวจสอบบริการไบโอมาร์ท [60] โดยใช้เวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ซึ่งมีความซับซ้อนกว่าหลายเท่าเป็นกรณีศึกษา จะกล่าวถึงกระบวนการหลังการตรวจสอบ ข้อเสนอแนะและสรุป

6.1 การตรวจสอบเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

จากการทดสอบการทำงานของเวิร์คโฟลว์ เพื่อการตรวจสอบบริการไบโอมาร์ทกับเวิร์คโฟลว์กรณีศึกษาอย่างง่ายในการค้นหา Gene ontology ในบทที่ 5 แล้วนั้น พบว่าเวิร์คโฟลว์ที่พัฒนาขึ้นมาสามารถทำงานได้ถูกต้อง โดยสามารถระบุฟิลเตอร์และแอ็ตตริบิวต์ที่ล้าสมัยและให้ข้อมูลที่สำคัญของฟิลเตอร์และแอ็ตตริบิวต์ที่ทันสมัยได้ ดังนั้นจึงตรวจสอบเวิร์คโฟลว์สำหรับการวิเคราะห์สนิปของมนุษย์ ซึ่งเป็นเวิร์คโฟลว์ที่ใช้มีบริการและการเชื่อมโยงข้อมูลที่ซับซ้อนและใช้ฐานข้อมูลที่หลากหลายดังที่กล่าวแล้วในบทที่ 4 และสามารถเข้าถึงได้ตามที่อยู่ URL ดังนี้ http://www.myexperiment.org/workflows/610 ขั้นตอนการตรวจสอบมีดังนี้

1) เปิดเวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ทด้วยโปรแกรมทาเวอร์นาดังรูปที่ 6.1



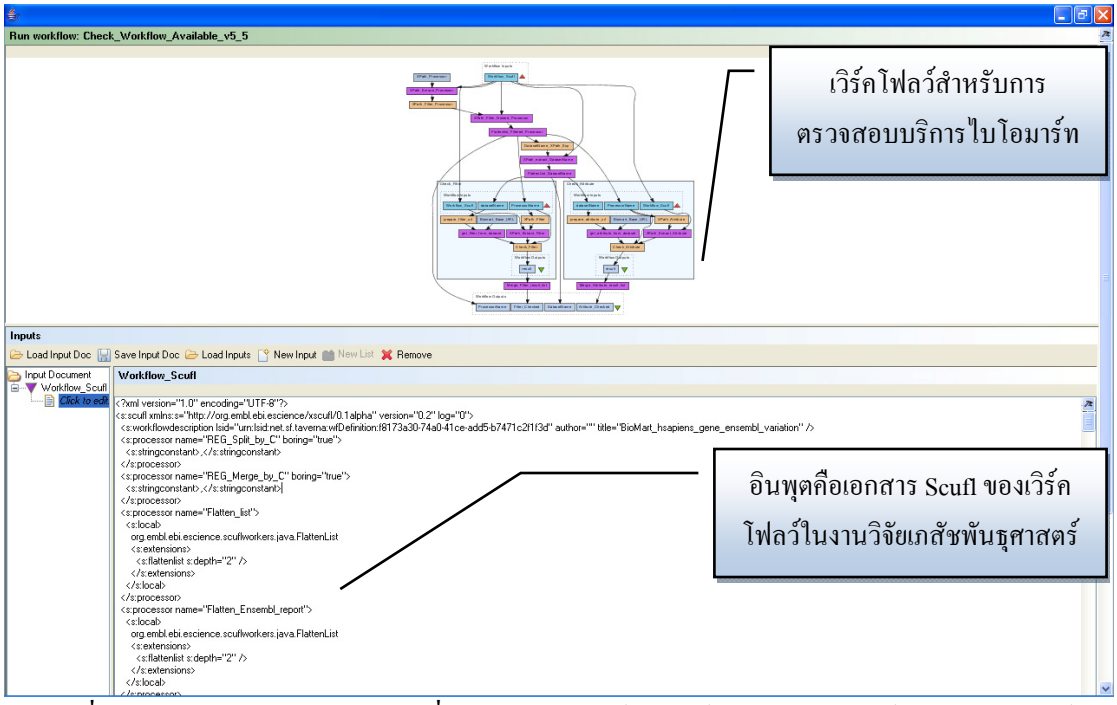
รูปที่ 6.1 เวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท

2) เริ่มการตรวจสอบบริการไบโอมาร์ท ในเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์โดยการสั่ง Run เวิร์คโฟลว์ แล้วจะได้ต่างหน้าดังรูปที่ 6.2 จะเห็นว่าหน้าต่างด้านบนแสดงเวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท และหน้าต่างด้านล่างเป็นพื้นที่สำหรับใส่อินพุตให้เวิร์คโฟลว์ทำงาน อินพุตในวิทยานิพนธ์นี้ก็คือ เอกสาร Scufl ของเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

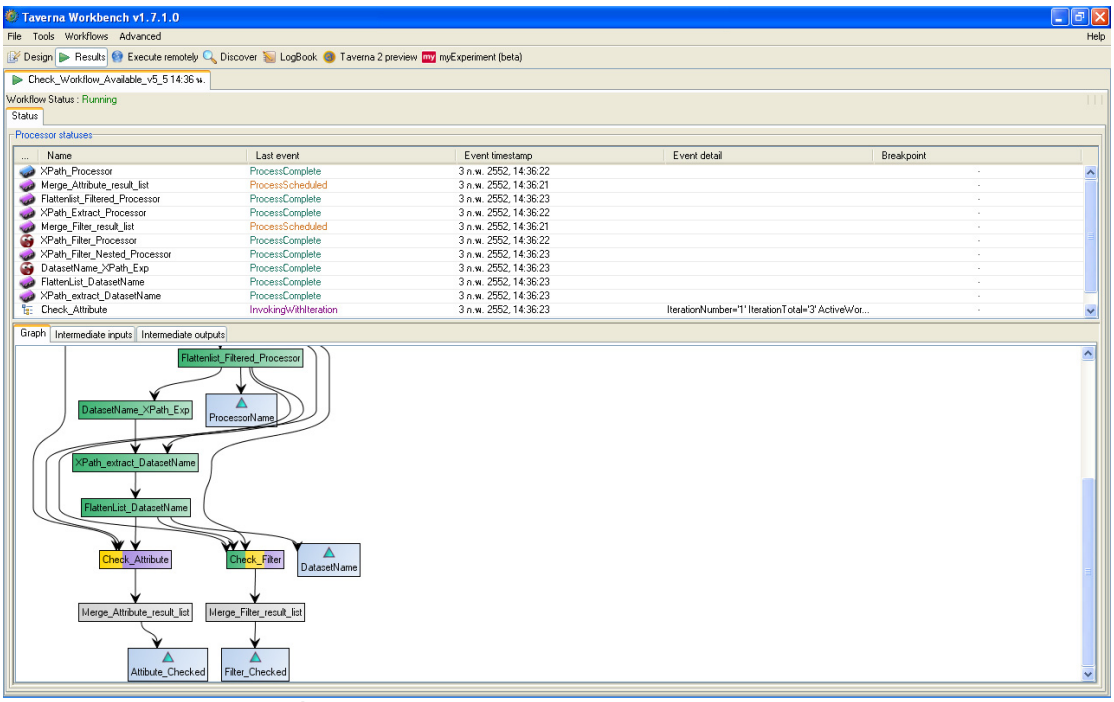
3) เมื่อใส่อินพุตพร้อมแล้วก็สามารถสั่ง Run เวิร์คโฟลว์ได้ รูปที่ 6.3 แสดงเวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องบริการไบโอมาร์ทกำลังทำงาน บล็อกที่มีสีเขียวแสดงสถานะการทำงานว่า บริการนี้ทำงานสำเร็จโดยไม่มีข้อผิดพลาด บล็อกที่มีสีม่วงแสดงสถานะการทำงานว่า บริการนี้กำลังทำงาน และบล็อกที่มีสีเหลืองบอกให้รู้ว่าบริการนี้มีการทำงานวนซ้ำหรือลูป จากรูปที่ 6.3 จะเห็นว่าคือบล็อกของบริการตรวจสอบฟิลเตอร์และแอ็ตตริบิวต์

4) เมื่อเวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้อง ของบริการไบโอมาร์ททำงานเสร็จสิ้น จะได้ผลลัพธ์เป็นรายงานการตรวจสอบฟิลเตอร์และแอ็ตตริบิวต์ รูปที่ 6.4 แสดงหน้าต่างรายงานผลการตรวจสอบฟิลเตอร์ของบริการไบโอมาร์ท โดยรายงานดังแสดงในรูปที่ 6.5 ซึ่งจะเห็นว่าฟิลเตอร์ทุกฟิลเตอร์ของบริการไบโอมาร์ทในเอกสาร Scufl อินพุตที่ใช้ในการตรวจสอบมีความทันสมัยและให้ข้อมูลรายละเอียดที่สำคัญของฟิลเตอร์นั้นๆ

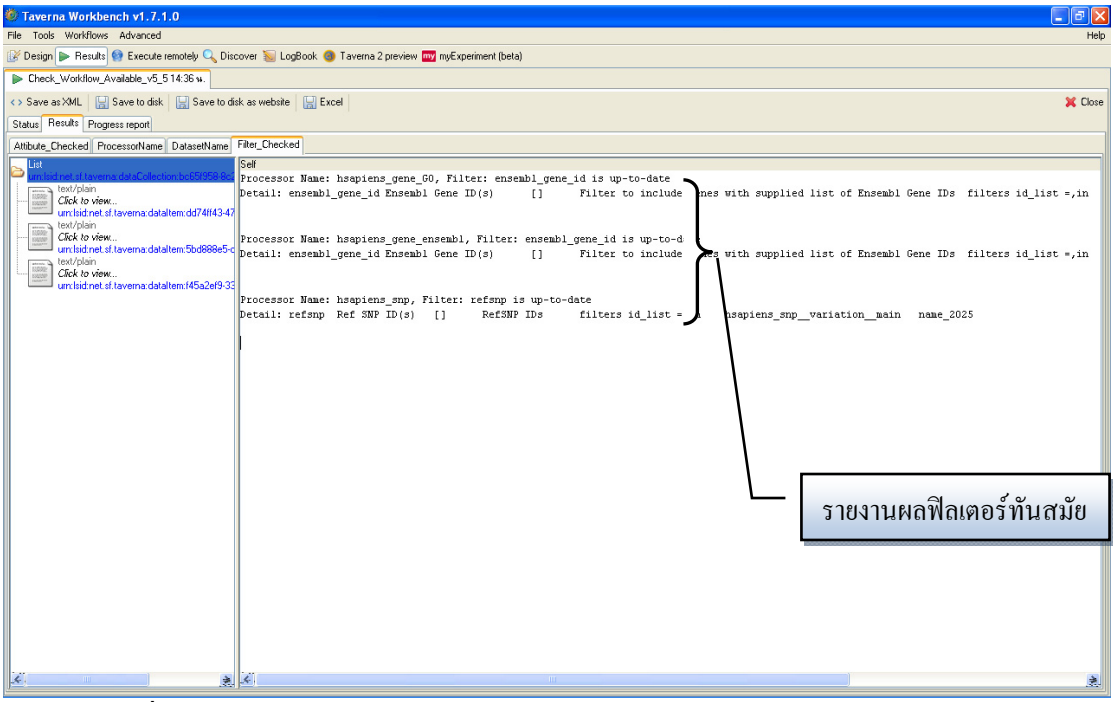
5) รูปที่ 6.6 แสดงหน้าต่างรายงานผลการตรวจสอบแอ็ตตริบิวต์ของบริการไบโอ-มาร์ท โดยรายงานแสดงดังรูปที่ 6.7 จะเห็นว่าแอ็ตตริบิวต์ทุกตัวของบริการไบโอมาร์ทล้าสมัยเสียแล้วและทำให้ไม่มีข้อมูลใดๆที่เกี่ยวข้องกับแอ็ตตริบิวต์ที่ล้าสมัยนั้นๆอยู่ที่บริการไบโอมาร์ทอีก



รูปที่ 6.2 หน้าต่างโปรแกรมพร้อมที่จะตรวจสอบเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์



รูปที่ 6.3 เวิร์คโฟลว์กำลังตรวจสอบบริการไบโอมาร์ท



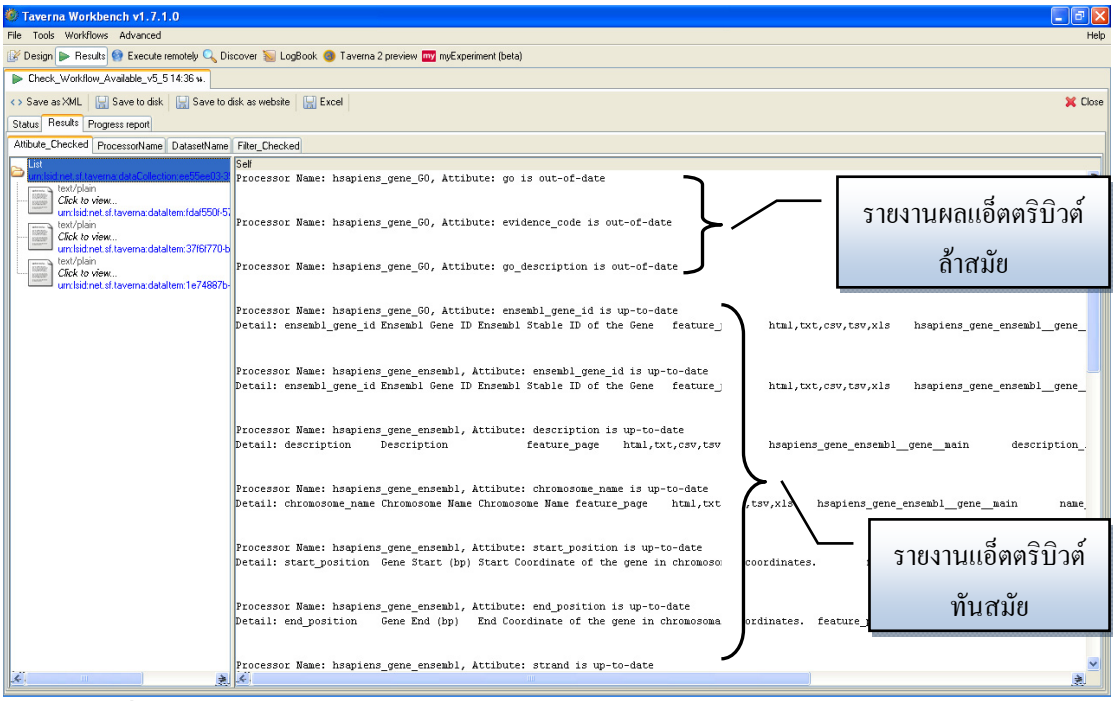
รูปที่ 6.4 ผลการตรวจสอบฟิลเตอร์ของบริการไบโอมาร์ทในโปรแกรมทาเวอร์นา

Processor Name: hsapiens_gene_GO, Filter: ensembl_gene_id is up-to-date
Detail: ensembl_gene_id Ensembl Gene ID(s) [] Filter to include genes with supplied list of Ensembl Gene IDs filters Id_list =,in hsapiens_gene_ensembl__gene__main stable_id_1023

Processor Name: hsapiens_gene_ensembl, Filter: ensembl_gene_id is up-to-date
Detail: ensembl_gene_id Ensembl Gene ID(s) [] Filter to include genes with supplied list of Ensembl Gene IDs filters id_list =,in hsapiens_gene_ensembl__gene__main stable_id_1023

Processor Name: hsapiens_snp, Filter: refsnp is up-to-date
Detail: refsnp Ref SNP ID(s) [] RefSNP IDs filters id_list =,in hsapiens_snp__variation__main name_1023

รูปที่ 6.5 รายงานการตรวจสอบฟิลเตอร์ของบริการไบโอมาร์ท



รูปที่ 6.6 ผลการตรวจสอบแอ็ตตริบิวต์ของบริการไบโอมาร์ทในโปรแกรมทาเวอร์นา

Processor Name: hsapiens_gene_GO, Attibute: go is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: evidence_code is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: go_description is out-of-date

รูปที่ 6.7 รายงานการตรวจสอบแอ็ตตริบิวต์ล้าสมัยของบริการไบโอมาร์ท

ตารางที่ 6.1 แสดงคุณสมบัติและผลการทดสอบเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ การทดสอบวัดผลการทดสอบ 10 ครั้ง เวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ท ใช้เวลาในการทำงานเฉลี่ย 26.5 วินาทีในการตรวจสอบบริการฟิลเตอร์ และแอ็ตตริบิวต์ได้อย่างถูกต้อง ซึ่งหากไม่ตรวจสอบก่อนจะใช้เวลานานถึง 14 ชั่วโมงกว่าจะรู้ว่าเวิร์คโฟลว์ดังกล่าวทำงานผิดพลาด

จากรูปที่ 6.4 จะเห็นว่าฟิลเตอร์ทุกตัวมีความทันสมัย โดยรายงานการตรวจสอบฟิลเตอร์ของบริการไบโอมาร์ทในรูปที่ 6.5 ซึ่งอธิบายข้อมูลตามที่ได้ไปดึงจากรีจิสทรีของบริการนั้นๆ ซึ่งข้อมูลเหล่านี้จะเป็นประโยชน์ในการอ้างอิงสำหรับการตรวจสอบในอนาคตต่อไป เพราะในอนาคตฟิลเตอร์เหล่านี้อาจจะถูกปรับปรุงได้เช่นกัน

จากรูปที่ 6.6 จะเห็นว่ามีแอ็ตตริบิวต์บางตัวที่ปรากฏในเอกสาร Scufl อินพุตได้ล้าสมัยไปแล้ว โดยในรายงานการตรวจสอบแสดงดังรูปที่ 6.7 พบว่าบริการที่ล้าสมัยไปแล้วก็คือบริการไบโอมาร์ทที่ทำหน้าที่ในการค้นหา Gene ontology ซึ่งเวิร์คโฟลว์ที่พัฒนาขึ้นนี้สามารถระบุได้ถูกต้องตรงตามสมมุติฐานและยังแสดงให้เห็นว่าเวิร์คโฟลว์สามารถตรวจสอบกับ Nested Workflow ได้อีกด้วย

ตารางที่ 6.1 คุณสมบัติและผลการทดสอบเวิร์คโฟลว์เภสัชพันธุศาสตร์

รายการ	จำนวน	หมายเหตุ
บริการทั้งหมด	55	-
ดาตาลิงค์ทั้งหมด	161	-
บริการที่เกี่ยวเนื่องกับบริการไบโอมาร์ท	17	เป็นบริการไบโอมาร์ท 3 บริการ
ดาตาลิงค์ที่เกี่ยวเนื่องกับบริการไบโอมาร์ท	70	-
ฟิลเตอร์	3	เป็นฟิลเตอร์ของบริการไบโอมาร์ท 3 บริการ
แอ็ตตริบิวต์	22	เป็นแอ็ตตริบิวต์ของบริการไบโอมาร์ท 3 บริการ
ข้อมูลอินพุต	1000	SNP IDs
เวลาทำงาน	14.00.50	ชั่วโมง
เวลาในการตรวจสอบ	26.5	วินาที / ทุกบริการทันสมัยยกเว้น Gene ontology

6.2 กระบวนการหลังการตรวจสอบ

เมื่อได้ตรวจสอบความถูกต้องของเวิร์คโฟลว์ และพบว่ามีบริการที่ล้าสมัยจึงจำเป็นต้องปรับปรุงแก้ไขให้เวิร์คโฟลว์มีความถูกต้องสมบูรณ์ พร้อมที่จะไปใช้งานที่เกี่ยวข้องกับงานวิจัยได้ตามความต้องการ ดังนั้นกระบวนการหลังการตรวจสอบจึงเป็นเรื่องที่จำเป็นเพื่อให้เวิร์คโฟลว์สามารถทำงานได้ถูกต้อง

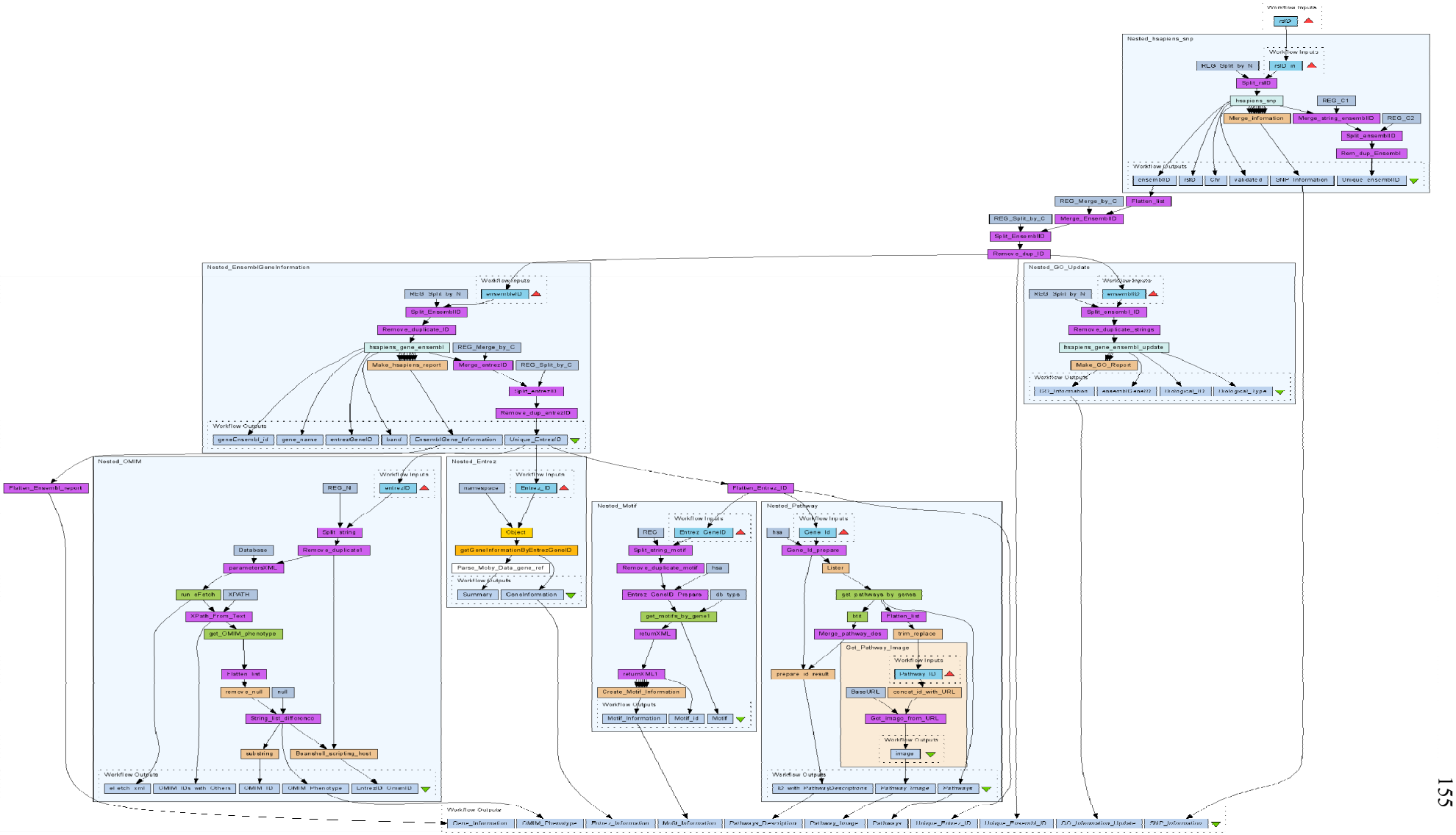
6.2.1 การปรับปรุงและการทดสอบซ้ำเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

จากผลการตรวจสอบ ทราบว่าแอ็ตตริบิวต์บางตัวของบริการไบโอมาร์ทชื่อ *hsapiens_gene_GO* ซึ่งมีการเรียกใช้ข้อมูลจากดาตาเซ็ต *hsapiens_gene_ensembl* ในการค้นหา

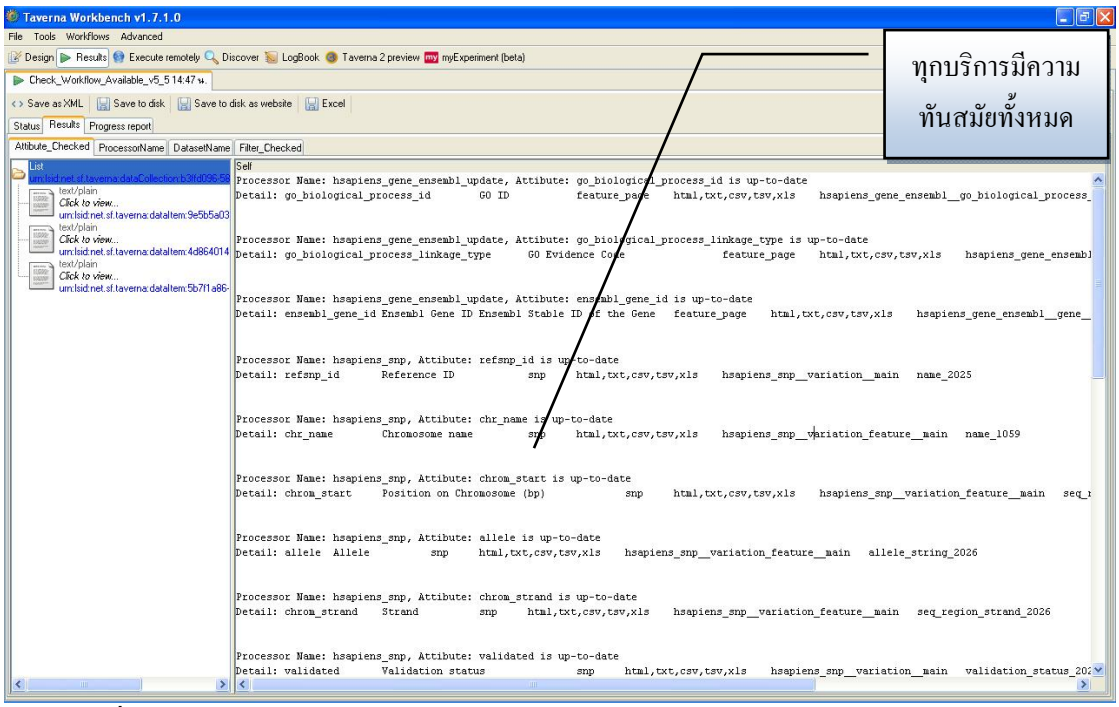
ข้อมูล Gene ontology เป็นบริการที่ล้าสมัยแล้ว สำหรับวิธีการปรับปรุงก็คือ จะต้องเอาบริการนี้ออกจากไดอะแกรมของเวิร์คโฟลว์แล้วทำการเพิ่มอันใหม่เข้ามา โดยอาจจะยังเป็นชื่อเดิมอยู่คือ *hsapiens_gene_ensembl* เพราะรายชื่อของบริการที่นำเสนอต่อผู้ใช้ใน ณ ขณะเวลาใดๆย่อมถือว่าทันสมัยที่สุดในขณะนั้น เมื่อเพิ่มบริการใหม่เข้ามาแล้วก็ปรับแต่งเลือกข้อมูลในส่วนที่ขาดหายไป ในกรณีนี้คือ Gene ontology และเชื่อมต่อการเชื่อมโยงข้อมูลกับพอร์ตอื่นๆต่อไป ในกรณีที่บริการที่เปลี่ยนใหม่แล้วนี้ ไม่มีฟิลเตอร์หรือแอ็ตตริบิวต์ที่ต้องการ ก็จำเป็นจะต้องค้นหาบริการอื่นๆที่ให้ผลลัพธ์ได้ตามความต้องการในหมวดหมู่ที่เกี่ยวข้อง

สำหรับเหตุผลที่ต้องเอาบริการที่ล้าสมัยออกและเพิ่มบริการใหม่มาแทนที่ ทั้งๆที่บริการเหล่านี้มีชื่อและดาตาเซ็ตเดียวกัน เนื่องจากการเข้าไปปรับแต่งบริการในหน้าต่างการปรับแต่งของบริการไบโอมาร์ทนั้นจะไม่ส่งผลกระทบใดๆ กับฟิลเตอร์และแอ็ตตริบิวต์ที่ล้าสมัยเลย เพราะว่าฟิลเตอร์และแอ็ตตริบิวต์เหล่านั้นได้ถูกปรับปรุงไปแล้ว โดยอาจจะถูกผู้ให้บริการเอาออกจากบริการไปเลยหรือเปลี่ยนไปเป็นชื่อใหม่ก็ได้แล้วแต่กรณี ดังนั้นในหน้าต่างการปรับแต่งจึงไม่สามารถยกเลิกการเลือกฟิลด์ข้อมูลเหล่านี้ได้ ทำให้ไม่มีผลกระทบต่อเอกสาร Scufl ของเวิร์คโฟลว์แต่อย่างใด รูปที่ 6.8 แสดงเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ที่ได้รับการแก้ไขเรียบร้อยแล้ว และสามารถเข้าถึงได้ทางเว็บไซต์ http://www.myexperiment.org/workflows/612

เมื่อแก้ไขเวิร์คโฟลว์เรียบร้อยแล้ว จากนั้นทำการตรวจสอบซ้ำอีกครั้งหนึ่ง ผลการทดสอบคือบริการไบโอมาร์ททุกบริการมีความทันสมัยทั้งหมดแล้ว และพร้อมทำงานได้ตามวัตถุประสงค์ ผลการตรวจสอบในโปรแกรมทาเวอร์นาแสดงดังรูปที่ 6.9 และตัวอย่างรายงานผลการตรวจสอบของแอ็ตตริบิวต์ซึ่งทันสมัยทุกแอ็ตตริบิวต์และให้ข้อมูลที่เกี่ยวข้องได้ถูกต้อง แสดงดังรูปที่ 6.10



รูปที่ 6.8 เวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ที่ได้รับการแก้ไขแล้ว



รูปที่ 6.9 การตรวจสอบเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์หลังการปรับแก้แล้ว

Processor Name: hsapiens_gene_ensembl_update, Attibute: go_biological_process_id is up-to-date
Detail: go_biological_process_id GO ID feature_page html,txt,csv,tsv,xls hsapiens_gene_ensembl__go_biological_process__dm dbprimary_acc_1074

Processor Name: hsapiens_gene_ensembl_update, Attibute: go_biological_process_linkage_type is up-to-date
Detail: go_biological_process_linkage_type GO Evidence Code feature_page html,txt,csv,tsv,xls hsapiens_gene_ensembl__go_biological_process__dm linkage_type_1024

Processor Name: hsapiens_gene_ensembl_update, Attibute: ensembl_gene_id is up-to-date
Detail: ensembl_gene_id Ensembl Gene ID Ensembl Stable ID of the Gene feature_page html,txt,csv,tsv,xls hsapiens_gene_ensembl__gene__main stable_id_1023

รูปที่ 6.10 ตัวอย่างรายงานผลการตรวจสอบของแอ็ตตริบิวต์ซึ่งมีความทันสมัยทุกตัว

6.2.2 การตรวจสอบความถูกต้องของผลลัพธ์

ผลลัพธ์ของเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ สามารถตรวจสอบความถูกต้องของผลลัพธ์กับงานวิจัยที่เกี่ยวข้องได้ [5][59][60] ปรากฏว่าเวิร์คโฟลว์สามารถค้นหาข้อมูลที่เกี่ยวข้องกับสนิปต่างๆที่สนใจได้ถูกต้อง และความถูกต้องของผลลัพธ์การทำงานได้รับการรับรองจากนักชีวสารสนเทศจากศูนย์วิจัยเภสัชพันธุศาสตร์และสารสนเทศ ของโรงพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล และศูนย์ความเป็นเลิศด้านชีววิทยาศาสตร์ของประเทศไทย [62]

6.3 ข้อเสนอแนะ

จากผลการทดลอบเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ บ่งบอกว่าจำเป็นต้องมีการปรับแต่ง หรือออกแบบเวิร์คโฟลว์ใหม่บ้างเล็กน้อย นั่นคือการเอาบริการที่ล้าสมัยออกและเพิ่มบริการใหม่เข้ามาแทนที่และจัดการเชื่อมโยงพอร์ตของข้อมูลต่างๆตามความต้องการ ในกรณีที่พบว่าแม้บริการจะมีหลายๆฟิลเตอร์ หากแต่มีฟิลเตอร์เพียงตัวเดียวที่ล้าสมัยไปแล้วก็จะทำให้ Data flow มีความผิดพลาดอยู่ เนื่องจากบริการจะต้องนำฟิลเตอร์ทั้งหมดเป็นไปเงื่อนไขของการคิวรี ดังนั้นฟิลเตอร์ทุกตัวจะต้องผ่านเงื่อนไขนั้น (AND condition) ซึ่งปัญหานี้จำเป็นต้องเปลี่ยนเป็นบริการใหม่เท่านั้นจึงจะสามารถทำงานต่อได้ ในกรณีของบริการที่มีเพียงแอ็ตตริบิวต์ก็เช่นเดียวกัน หากเพียงแอ็ตตริบิวต์เดียวที่ล้าสมัยไปแล้วก็ส่งผลให้ Data flow มีความผิดพลาด ถึงแม้ว่าแอ็ตตริบิวต์ตัวนั้นจะไม่ได้นำไปใช้ประมวลผลกับกระบวนการถัดไปก็ตาม

ฟิลเตอร์และแอ็ตตริบิวต์ที่ล้าสมัย อันเนื่องมาจากการปรับปรุงของบริการไบโอมาร์ทเหล่านี้ จะไม่แสดงตัวที่หน้าต่างการปรับแต่งของบริการในโปรแกรมทาเวอร์นา ดังนั้นผู้ใช้งานจึงไม่ทราบความล้าสมัย และไม่สามารถยกเลิกการเลือกฟิลเตอร์และแอ็ตตริบิวต์เหล่านี้ได้ ผู้ใช้ที่ชำนาญอาจจะแก้เอกสาร Scufl ของเวิร์คโฟลว์ด้วยตัวเองได้ แต่วิธีนี้ก็สร้างความสับสนและทำให้เกิดข้อผิดพลาดได้ง่าย

ในที่สุดแล้ว คำแนะนำก็คือการเพิ่มบริการใหม่ โดยยังเป็นชื่อบริการเดิมอยู่ซึ่งปรากฏอยู่ในรายงานการตรวจสอบเวิร์คโฟลว์ของระบบตรวจสอบ ที่นำเสนอในวิทยานิพนธ์นี้ ผู้ใช้งานสามารถเลือกบริการนี้ใหม่อีกครั้งจากหน้าต่าง *Service available panel* ในโปรแกรมทาเวอร์นา จากนั้นเชื่อมต่อกับบริการและเชื่อมโยงข้อมูลที่เกี่ยวข้องเข้าด้วยกัน และในกรณีที่

บริการใหม่ที่เพิ่มเข้ามานี้ไม่มีฟิลเตอร์หรือแอ็ตตริบิวต์ที่ต้องการ จึงมีความจำเป็นต้องเลือกบริการอื่นๆที่อยู่ในหมวดหมู่เดียวกันมาใช้งานแทน

งานวิจัยในวิทยานิพนธ์นี้ สร้างเวิร์คโฟลว์สำหรับการตรวจสอบฟิลด์ข้อมูลเงื่อนไข และฟิลด์ข้อมูลผลลัพธ์ของบริการไบโอมาร์ทในสิ่งแวดล้อมของโปรแกรมทาเวอร์นา เวิร์คโฟลว์สำหรับตรวจสอบบริการไบโอมาร์ทนี้ ช่วยในการปรับปรุงกระบวนการของเวิร์คโฟลว์ โดยไม่ต้องรอให้เวิร์คโฟลว์ทำงานเสร็จเสียก่อนที่จะทราบว่าเวิร์คโฟลว์ไม่สมบูรณ์ และไม่สามารถให้ผลลัพธ์ที่ต้องการได้ โดยสามารถประหยัดเวลาในการทำงานได้มากโดยไม่ต้องเสียไปกับการทำงานที่ไม่ได้ผลลัพธ์ใดๆ หรือได้ผลลัพธ์ที่ไม่ต้องการและสามารถลดเวลาในการดีบักเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์ลงได้มาก นอกจากนี้เวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ทยังนำเสนอข้อมูลที่เกี่ยวข้องกับบริการที่ทันสมัยนั้นๆสำหรับการตรวจสอบในอนาคตอีกด้วย

6.4 สรุป

ในบทนี้ ได้กล่าวถึงการทดสอบเวิร์คโฟลว์สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ท โดยใช้เวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์เป็นกรณีศึกษา รายงานผลการตรวจสอบเวิร์คโฟลว์ สามารถระบุฟิลเตอร์และแอ็ตตริบิวต์ที่ล้าสมัยได้อย่างถูกต้อง หรือกล่าวอีกนัยหนึ่งก็คือ เวิร์คโฟลว์สำหรับการตรวจสอบสามารถระบุบริการไบโอมาร์ทที่ล้าสมัยได้อย่างถูกต้อง ซึ่งใช้เวลาน้อยกว่าครึ่งนาทีและสามารถลดเวลาในการดีบักเวิร์คโฟลว์ลงได้อย่างมีนัยสำคัญ รวมทั้งให้ข้อมูลที่สำคัญในการช่วยปรับแก้เวิร์คโฟลว์ได้ โดยการลบบริการเดิมและเพิ่มบริการใหม่ลงไป จากนั้นเชื่อมต่อการเชื่อมโยงข้อมูลกับบริการอื่นๆและทดสอบซ้ำ ซึ่งเวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์สามารถทำงานได้ถูกต้องโดยปราศจากข้อผิดพลาดใดๆ

บทที่ 7

สรุป

ในบทนี้จะกล่าวสรุปผลการวิจัยที่ได้ดำเนินการสำหรับวิทยานิพนธ์นี้ รวมทั้งข้อเสนอแนะต่างๆ เพื่อเป็นประโยชน์ต่อการทำวิจัยเกี่ยวกับเทคโนโลยีมายกริดและโปรแกรมทาเวอร์นาต่อไป

7.1 ภาพรวมของงานวิทยานิพนธ์

วิทยานิพนธ์นี้เป็นการประยุกต์ใช้เทคโนโลยีมายกริด โปรแกรมทาเวอร์นา และเว็บเซอร์วิสเพื่อแก้ไขปัญหาในงานวิจัยทางด้านชีวสารสนเทศศาสตร์ ซึ่งเป็นงานวิจัยที่ทำร่วมกับหน่วยงานวิจัยที่มีความชำนาญทางด้านชีวสารสนเทศศาสตร์ของมหาวิทยาลัยสงขลานครินทร์ และโรงพยาบาลรามาธิบดี คณะแพทยศาสตร์ มหาวิทยาลัยมหิดล วิทยานิพนธ์นี้ได้บรรลุวัตถุประสงค์ดังต่อไปนี้

1) ได้นำเสนอวิธีการพัฒนาเวิร์คโฟลว์ และสร้างการเชื่อมต่อในเวิร์คโฟลว์สำหรับแก้ไขปัญหาในงานวิจัย การทำนายผลกระทบของสนิปของมนุษย์และการวิเคราะห์สนิปของกุ้ง โดยใช้เทคโนโลยีมายกริด, โปรแกรมทาเวอร์นา, เว็บเซอร์วิสและใช้เวิร์คโฟลว์ทั้งสองนี้เป็นกรณีศึกษาของงานวิทยานิพนธ์

2) ได้นำเสนอกลไกในการลดเวลาในการพัฒนา และทดสอบเวิร์คโฟลว์โดยพัฒนาการตรวจสอบความถูกต้องของเวิร์คโฟลว์ก่อนการทำงานจริง โดยจะมุ่งเน้นไปที่การตรวจสอบบริการไบโอมาร์ทภายใต้สิ่งแวดล้อมของโปรแกรมทาเวอร์นา ที่สามารถตรวจสอบคุณสมบัติของบริการไบโอมาร์ทได้ว่าทันสมัยพร้อมทำงานหรือล้าสมัยไปแล้ว ซึ่งสามารถระบุบริการที่ล้าสมัยไปแล้วในเวิร์คโฟลว์ และให้ข้อมูลที่มีนัยสำคัญต่อผู้ใช้งานจะได้มีความสะดวกและง่ายต่อการปรับแก้เวิร์คโฟลว์ซึ่งสามารถลดเวลาที่เสียไปได้อย่างมาก

3) ได้นำเสนอกลไกในการลดปัญหาการเกินเวลา (Time-out) และความไม่คงเส้นคงวาในการทำงาน (Inconsistancy) ที่ส่งผลให้เวิร์คโฟลว์ทำงานไม่สำเร็จโดยพัฒนาบริการท้องถิ่นขึ้นมาใช้งานเอง (Local web services)

วิทยานิพนธ์นี้บรรลุวัตถุประสงค์ทั้ง 3 ข้อดังกล่าว และจะสรุปผลลัพธ์ของแต่ละงานในวิทยานิพนธ์ดังนี้

7.2 การเลือกบริการที่เหมาะสมกับลักษณะของงานและการพัฒนาเวิร์คโฟลว์

งานวิทยานิพนธ์นี้ได้นำเสนอการเลือกบริการที่เหมาะสม และการวาดเวิร์คโฟลว์โดยให้ข้อเสนอแนะที่เป็นกระบวนการทำงานแก่ผู้ใช้งาน โดยแบ่งวิธีการค้นหาและการเลือกบริการออกเป็น 2 วิธีคือ

- การค้นหาบริการจากเว็บพอร์ทอลของมายกริดซึ่งเรียกว่า Biological Web Service โดยสามารถเข้าถึงได้ที่ URL ดังนี้ http://www.mygrid.org.uk/wiki/Mygrid/BiologicalWebServices
- การค้นหาบริการจากอินเตอร์เน็ตโดยปลักอินของโปรแกรมทาเวอร์นาเอง หรือ feta ซึ่งเป็นการค้นหาเว็บเซอร์วิสโดยใช้พื้นฐานจากคำอธิบายของหน้าที่ในการทำงานของบริการนั้นๆ (Description) ด้วย Service ontology ซึ่งจะเก็บอยู่ที่รีจิสทรีของบริการหรือ UDDI ของมายกริด หากค้นหาบริการที่ต้องการได้ก็สามารถทดสอบการใช้งานจากโปรแกรมทาเวอร์นาได้ทันที

นอกจากนี้ยังนำเสนอวิธีการพัฒนาหรือการวาดเวิร์คโฟลว์ โดยเริ่มต้นที่สมมุติฐานที่ว่าผู้วิจัยได้การออกแบบ Flowchart หรือ Data flow diagram ของระบบงานวิจัยเสร็จสิ้นแล้ว ผู้วิจัยมองเห็นอินพุต การไหลของข้อมูลและเอาท์พุตโดยรวมจากการออกแบบได้ โดยพร้อมที่จะพัฒนาเวิร์คโฟลว์ของระบบงานตามที่ได้ออกแบบ

7.3 บริการท้องถิ่นในการทำนายโครงสร้างต้นสายวิวัฒนาการของกุ้ง

ในการสร้างเวิร์คโฟลว์เพื่อวิเคราะห์สนิปในกุ้ง ซึ่งเป็นกรณีศึกษาเบื้องต้นวิทยานิพนธ์นี้ แต่เวิร์คโฟลว์ที่พัฒนาครั้งแรกไม่สามารถทำงานได้สำเร็จเนื่องจากเกิดปัญหาการเกิน

เวลาและความไม่คงเส้นคงวาในการทำงานระหว่างการถ่ายโอนข้อมูลระหว่างเว็บเซอร์วิสของ EBI ในส่วนของการสร้างโครงสร้างต้นไม้สายวิวัฒนาการของกุ้ง

วิทยานิพนธ์นี้ได้พัฒนาเว็บเซอร์วิสท้องถิ่นหรือบริการท้องถิ่น (Local web services) ขึ้นมาใช้งานเองภายในระบบเวิร์คโฟลว์ โดยใช้ Soaplab Analysis Tool เป็นเครื่องมือช่วยสร้างตัวเชื่อมประสานหรือ Web service interfaces ให้กับโปรแกรมประยุกต์ต่างๆที่เกิดปัญหาการเกินเวลา เพื่อให้โปรแกรมทาเวอร์นาเข้าถึงได้ และสร้าง Java Web Service (JWS) ในการจัดการอินพุตเอาท์พุตภายในเวิร์คโฟลว์ด้วย บริการท้องถิ่นที่สร้างขึ้นสามารถแก้ไขปัญหาการเกินเวลาและความไม่คงเส้นคงวาในการทำงานได้เป็นอย่างดี สามารถลดเวลาการทำงานลงได้อย่างมีนัยสำคัญเมื่อเปรียบเทียบกับการทำงานด้วยวิธีการแบบเดิม (คัดลอกแล้ววาง) และผลลัพธ์ที่ได้จากการทำงานของเวิร์คโฟลว์เพื่อวิเคราะห์สนิปในกุ้งนี้ ได้รับการตรวจสอบความถูกต้องจากศูนย์วิจัยจีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์

7.4 เวิร์คโฟลว์สำหรับวิเคราะห์สนิปของมนุษย์

วิทยานิพนธ์นี้ได้กล่าวถึงการพัฒนาวิธีการทำนายผลกระทบ อันเนื่องมาจากการเปลี่ยนแปลงสนิปต่างๆของมนุษย์ รวมทั้งสร้างระบบที่มีความสามารถในการคัดเลือกตำแหน่งสนิปที่น่าสนใจในการนำไปศึกษาต่อ ทางด้านหน้าที่การทำงานของสนิปต่างๆต่อไปด้วยเวิร์คโฟลว์ การแก้ปัญหางานวิจัยใช้โปรแกรมทาเวอร์นา ที่มีความสามารถในการสร้างและจัดการรูปแบบข้อมูล รวมทั้งมีความสามารถในการดึงข้อมูลจากแหล่งต่างๆ โดยประสานงานกับเว็บเซอร์วิสชนิดต่างๆและแสดงผลออกมาในรูปแบบที่ง่ายต่อการนำไปศึกษา หรือใช้งานต่อได้และได้รับการตรวจสอบจากนักชีวสารสนเทศแล้วว่าทำงานได้ถูกต้องตรงตามความต้องการ และเร็วกว่าการทำงานตามปกติ และใช้เวิร์คโฟลว์นี้เป็นกรณีศึกษาหลักของวิทยานิพนธ์

7.5 เวิร์คโฟลว์สำหรับการตรวจสอบบริการไบโอมาร์ท

วิทยานิพนธ์นี้นำเสนอแนวคิดในการออกแบบ การพัฒนา และการทดสอบเวิร์คโฟลว์สำหรับตรวจสอบการเปลี่ยนแปลง ของบริการไบโอมาร์ทในเวิร์คโฟลว์วิเคราะห์ของมนุษย์ของงานวิจัยเภสัชพันธุศาสตร์แบบอัตโนมัติ และเวิร์คโฟลว์ที่พัฒนาขึ้นใช้บริการท้องถิ่นที่

โปรแกรมทาเวอร์นาเตรียมไว้ให้สำหรับการสร้างสคริปต์ของภาษา Java ขึ้นมาทำงานตามอัลกอริทึมที่ออกแบบไว้ จากการทดสอบด้วยเวิร์คโฟลว์กรณีศึกษาอย่างง่ายพบว่า เวิร์คโฟลว์สำหรับตรวจสอบการเปลี่ยนแปลงของบริการไบโอมาร์ทซึ่งสามารถระบุได้ว่า บริการไบโอมาร์ทนั้นๆได้ล้าสมัยไปแล้วได้อย่างถูกต้อง

จากนั้นได้นำเสนอผลการทดสอบของเวิร์คโฟลว์ สำหรับการตรวจสอบความถูกต้องของบริการไบโอมาร์ท โดยมีเวิร์คโฟลว์วิเคราะห์สนิปของมนุษย์เป็นกรณีศึกษา ซึ่งสามารถระบุบริการไบโอมาร์ทที่ล้าสมัยได้อย่างถูกต้องโดยใช้เวลาน้อยกว่าครึ่งนาที และรายงานผลการตรวจสอบ ยังสามารถลดเวลาในการดีบักเวิร์คโฟลว์ลงได้อย่างมีนัยสำคัญ ซึ่งให้ข้อมูลที่สำคัญในการช่วยปรับแก้เวิร์คโฟลว์ได้ และทดสอบซ้ำหลังจากปรับแก้แล้ว เวิร์คโฟลว์กรณีศึกษาสามารถทำงานได้ถูกต้องโดยปราศจากข้อผิดพลาดใดๆ และความถูกต้องของผลลัพธ์การทำงานได้รับการรับรองจากนักชีวสารสนเทศ จากศูนย์วิจัยเภสัชพันธุศาสตร์และสารสนเทศของโรงพยาบาลรามาธิบดี คณะแพทยศาสตร์ มหาวิทยาลัยมหิดล และศูนย์ความเป็นเลิศด้านชีววิทยาศาสตร์ของประเทศไทย

7.6 อุปสรรคและปัญหา

- การพัฒนาระบบงานแบบเวิร์คโฟลว์ในโปรแกรมทาเวอร์นา ยังถือว่าเป็นงานวิจัยเฉพาะด้านและต้องเป็นไปตามมาตรฐานที่โครงการมายกริดนำเสนอ เช่น สถาปัตยกรรมของ Soaplab หรือภาษา Scufl ของเอกสารเวิร์คโฟลว์ ซึ่งผู้วิจัยต้องทำความรู้จักกับมาตรฐานใหม่นี้ ทั้งนี้โครงการมายกริดก็ไม่ได้สนับสนุนการใช้ภาษา Scufl ภายนอกสิ่งแวดล้อมของโปรแกรมทาเวอร์นา ภาษา Scufl เป็นภาษาที่เฉพาะเจาะจงสำหรับระบบงานแบบเวิร์คโฟลว์และประยุกต์ใช้กับมาตรฐานอื่นๆได้ไม่ง่ายดายนัก
- โปรแกรมทาเวอร์นาอนุญาตให้ผู้วิจัยสามารถสร้างอัลกอริทึม สำหรับการทำงานได้ตามความต้องการ บริการท้องถิ่นบีนเชลล์ที่โปรแกรมนำเสนอต่อผู้วิจัยนั้นรองรับการโปรแกรมภาษา Java ดังนั้นผู้วิจัยจะต้องเขียนภาษาสคริปต์ของภาษา Java เท่านั้น ปัญหาที่เกิดขึ้นคือ หากอัลกอริทึมมีกระบวนการทำงานที่ซับซ้อน จะทำให้การดีบักความถูกต้องของสคริปต์ทำได้ยาก เนื่องจากจะต้องสั่งให้เวิร์คโฟลว์ทำงานจริงๆจึงจะสามารถดีบักความถูกต้องของสคริปต์ได้

7.7 ข้อเสนอแนะ

- การพัฒนาเว็บเซอร์วิสท้องถิ่นหรือบริการท้องถิ่น (Local web services) ขึ้นมาใช้งานเองภายในระบบเวิร์คโฟลว์เพื่อวิเคราะห์สนิปในกุ้ง เป็นบริการที่ทำงานบนโครงสร้างพื้นฐานของระบบคอมพิวเตอร์คลัสเตอร์ แต่เนื่องจากชุดแอ็พพลิเคชัน Emboss และ PHYLIB ที่ทางศูนย์จีโนมิกส์และชีวสารสนเทศแห่งมหาวิทยาลัยสงขลานครินทร์มีใช้งาน เป็นโปรแกรมในการทำนายโครงสร้างสายวิวัฒนาการยังคงเป็นแอ็พพลิเคชันแบบ Sequencial ดังนั้นบริการท้องถิ่นที่ได้พัฒนาในวิทยานิพนธ์นี้ จึงยังไม่สามารถดึงความสามารถด้านการประมวลผลแบบขนานของระบบคอมพิวเตอร์คลัสเตอร์มาใช้งานได้เต็มที่ บริการท้องถิ่นที่พัฒนาทำงานเพียงโหนดเดียวเท่านั้นคือที่โหนด front end ของระบบคลัสเตอร์ Hanuman ของศูนย์กริดมหาวิทยาลัยสงขลานครินทร์ ข้อเสนอแนะคือ นอกจากจะมองว่าบริการแต่ละบริการในเวิร์คโฟลว์จะทำงานเป็นลักษณะการกระจายแล้ว บริการท้องถิ่นที่พัฒนาก็อาจจะปรับปรุงให้ทำงานแบบขนานได้ ซึ่งต้องใช้ชุดแอ็พพลิเคชัน Emboss และ PHYLIB ที่ทำงานแบบขนานสำหรับงานบนระบบคอมพิวเตอร์คลัสเตอร์ ซึ่ง ณ ปัจจุบันที่ทำวิทยานิพนธ์นี้ยังไม่มีชุดแอ็พพลิเคชัน Emboss และ PHYLIB ที่ทำงานแบบขนานสำหรับงานบนระบบคอมพิวเตอร์คลัสเตอร์ออกมาให้ใช้งาน อย่างไรก็ตาม ยังมีงานวิจัยที่พัฒนาระบบการกระจายงานในระบบขนานสำหรับชุดแอ็พพลิเคชัน Emboss ขึ้นมาใช้งานเอง [63] นอกจากนี้สถาบัน National Institutes of Health (NIH) ซึ่งเป็นผู้ให้บริการระบบคอมพิวเตอร์คลัสเตอร์ Biowulf [64] พัฒนาโปรแกรมชื่อ Swarm [65] สำหรับช่วยกระจายงานให้เป็นแบบขนานบนระบบคอมพิวเตอร์คลัสเตอร์ได้ และนักวิจัยที่สนใจสามารถดาวน์โหลดโปรแกรม Swarm มาใช้งานกับระบบคอมพิวเตอร์คลัสเตอร์ของตนเองได้อีกด้วย แต่ทั้งนี้การทำงานบนระบบคอมพิวเตอร์คลัสเตอร์ ซึ่งมีโครงสร้างพื้นฐานที่เป็นฮาร์ดแวร์ที่ทรงประสิทธิภาพย่อมทำให้บริการท้องถิ่นมีความน่าเชื่อถือในการทำงานและมีความสามารถในเข้าถึงสูง (Accessibility)

- การเสนอแนวคิดในการออกแบบ การพัฒนา และการทดสอบเวิร์คโฟลว์สำหรับตรวจสอบการเปลี่ยนแปลงของบริการ ที่ใช้ในสำหรับวิเคราะห์สนิปของมนุษย์แบบอัตโนมัติ เป็นการตรวจสอบกับบริการไบโอมาร์ทเท่านั้น เนื่องจากเป็นบริการหลักที่ใช้ในเวิร์คโฟลว์กรณีศึกษา ข้อเสนอแนะคือ การพัฒนาระบบตรวจสอบนี้ให้สามารถตรวจสอบกับบริการอื่นๆได้ด้วย เช่น BioMoby หรือ e-Utilities ของ NCBI เป็นต้น ในขั้นตอนการปรับแก้เวิร์คโฟลว์หลังการตรวจสอบก็อาจจะนำเสนอกลไกในการแก้ไขเวิร์คโฟลว์แบบอัตโนมัติได้ นอกนี้กลไกตรวจสอบเหล่านี้ ก็อาจจะไม่จำเป็นต้องอยู่ในรูปแบบของเวิร์คโฟลว์เสมอไปก็ได้ เช่น อาจจะอยู่ในรูปแบบของเว็บเซอร์วิส เป็นต้น

เอกสารอ้างอิง

[1] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M.,Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A. and Li, P., "Taverna: a tool for the composition and enactment of bioinformatics workflows," BIOINFORMATICS, 2004 Vol. 20, no. 17, pp. 3045–3054, doi:10.1093/bioinformatics/bth361.

[2] Wolstencroft, K.; Oinn, T.; Goble, C.; Ferris, J.; Wroe, C.; Lord, P.; Glover, K.; Stevens, R., "Panoply of utilities in Taverna," First International Conference on e-Science and Grid Computring, 5-8 Dec. 2005.

[3] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Peter Li, P. and Oinn, T., "Taverna: a tool for build and running workflows of services," Nucleic Acids Research, 2006, Vol. 34, Web Server issue W729–W732.

[4] Turi, D., Missier, P., Goble, C., De Roure, D., Oinn, T., "Taverna Workflows: Syntax and Semantics, e-Science and Grid Computing," IEEE International Conference on 10-13 Dec. 2007, pp. 441–448.

[5] Wellcome Trust Case Control Consortium, "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," Nature 2007, 447, pp. 661-678.

[6] SNP. 2009. http://www.ncbi.nlm.nih.gov/sites/entrez?db=snp.

[7] Young C. Song, Edward Kawas, Ben M. Good , Mark D. Wilkinson and Scott J. Tebbutt, "DataBiNS: a BioMoby-based data-mining workflow for biological pathways and non-synonymous SNPs," Bioinformatics Advance Access originally published online on January 18, 2007, Bioinformatics 2007 23(6), pp. 780-782, doi:10.1093/bioinformatics/btl648.

[8] Taverna project website. 2009. http://taverna.sourceforge.net.

[9] myGrid. 2009. http://www.mygrid.org.uk/.

[10] R. Stevens, H.J. Tipney, C. Wroe, T. Oinn, M. Senger, P. Lord, C.A. Goble, A. Brass and M. Tassabehji, "Exploring Williams-Beuren Syndrome Using myGrid in Proceedings of 12th International Conference on Intelligent Systems in Molecular Biology," 31st Jul-4th Aug 2004, Glasgow, UK, published Bioinformatics Vol. 20 Suppl. 1 2004, i303-i310.

[11] BioMart. 2009. http://www.biomart.org/.

[12]Damian Smedley, Syed Haider, Benoit Ballester, Richard Holland, Darin London, Gudmundur Thorisson and Arek Kasprzyk., "BioMart - biological queries made easy," BMC Genomics 2009, 10:22doi:10.1186/1471-2164-10-22, Published 14 January 2009, http://www.biomedcentral.com/1471-2164/10/22.

[13]Wanna, W., Rolland, J.L., Bonhomme, F. and Phongdara, A., “Population genetic structure of Penaeus merguiensis in Thailand based on nuclear DNA variation,” J. Exp. Mar. Bio, 2004, Ecol. 311, pp. 63-78.

[14]ธีราวุฒิ ภู่สันติสัมพันธ์. 2549. การศึกษา Single nucleotide polymorphisms ของยีน amylase ในกุ้งแชบ๊วย. วิทยานิพนธ์วิทยาศาสตรบัณฑิต, สาขาวิชาชีวเคมี คณะวิทยาศาสตร์, มหาวิทยาลัยสงขลานครินทร์.

[15]Phylogenetic trees. 2009. http:// cnx .org /content /m11052 /latest/.

[16]BioDAS. 2009. http://www.biodas.org/wiki/Main_Page.

[17]Dasty2, a protein DAS client. 2009. http://www.ebi.ac.uk/dasty/.

[18]MySQL. 2009. www.mysql.com/.

[19]Oracle. 2009. http://www.oracle.com/index.html.

[20]PostgreSQL. 2009. www.postgresql.org/.

[21]GNU Lesser General Public Licens, GNU Project - Free Software. 2009. http://www.gnu.org /licenses/lgpl.html.

[22]Peter M. Rice, Alan J. Bleasby, Syed A. Haider, Jon C. Ison, Shaun McGlinchey, Mahmut Uludag, "EMBRACE: Bioinformatics Data and Analysis Tool Services for e-Science," e-science,pp.146, Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), 2006, DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/E-SCIENCE.2006.57.

[23]Java Database Connectivity (JDBC), Java SE Technology, Database. 2009. http://java.sun. com /javase/technologies/database/

[24]Nevirapine, FDA Public Health Advisory for Nevirapine (Viramune). 2009. http://www.fda. gov/CDER/drug/advisory/nevirapine.htm.

[25]Gold, S., "Stevens-Johnson Syndrome," Br Med J. 1953 May 16; 1(4819): 1111. PMCID: PMC2016447. Available. 2009. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid =2016447.

[26]Lude Franke, Harm van Bakel, Like Fokkens, Edwin D. de Jong, Michael Egmont-Petersen, Cisca Wijmenga, “Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes.” Am J Hum Genet 2006 Jun; 78(6):1011-25.

[27]dbSNP Overview, NCBI. 2009. http://www.ncbi.nlm.nih.gov/projects/SNP/get_html.cgi?whichHtml=overview.

[28]KEGG: Kyoto Encyclopedia of Genes and Genomes. 2009. http://www.genome.jp/kegg/.

[29]Lei Bao, Mi Zhou and Yan Cui, "nsSNPAnalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms," Nucleic Acids Research 2005 33(Web Server Issue):W480-W482; doi:10.1093/nar/gki372.

[30]Karchin, R., Diekhans, M., Kelly, L., Thomas, DJ., Pieper, U, Eswar, N., Haussler, D. and Sali, A., "LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources," Bioinformatics. 2005 Jun 15;21(12):2814-20. Epub 2005 Apr 12.

[31]ศิวดล ไชยศิริ, ศิริชัยโจรน์วิภาค และผศ.ดร. ภุชงค์ อุทโยภาศ, กริดเซอร์วิส – เว็บเซอร์วิสสายพันธุ์ใหม่. 2009. http://apstf.cpe.ku.ac.th/oldweb/download/GridService.pdf.

[32]Web Service articles. 2009. http://www.service-architecture.com/web-services/articles/.

[33]SOAP Specifications. 2009. http://www.w3.org/TR/soap/.

[34]Web Service Definition Language (WSDL). 2009. www.w3.org/TR/wsdl.

[35]PSU-Grid. 2009. http://psu-grid.coe.psu.ac.th/.

[36]FASTA format description. 2009. http:// www.ncbi.nlm.nih.gov/blast/fasta.shtml.

[37]EBI Web Service for EMBOSS-4.1 application. 2009. http://www.ebi.ac.uk/soaplab/emboss4/services.

[38]Damkliang, K., Tandayya, P., Sangket, U., and Phusantisampan, T. 2009. “Workflow and Supporting Services for Single Nucleotide Polymorphisms Analysis Gene Using Taverna.” Proceedings of 2009 International Conference on Information Management and Engineering (ICIME 2009), Kuala Lumpur, Malaysia, April 3-5, 2009. pp. 27-31.

[39]European Bioinformatics Institute. 2009. www.ebi.ac.uk/

[40]Web API for Bioinformatics. 2009. http://xml.ddbj.nig.ac.jp/index.html

[41]NCBI BLAST. 2009. www.ncbi.nlm.nih.gov/BLAST/

[42]EBI Tools, InterProScan. 2009. www.ebi.ac.uk/Tools/InterProScan/

[43]SeqVISTA: Towards an Integrative Plantform for BioSeqs Analyses. 2009. http://zlab.bu.edu/SeqVISTA/.

[44]TreeView X. 2009. http://darwin.zoology.gla.ac.uk/~rpage/treeviewx/index.html.

[45]Web Services - Axis. 2009. http://ws.apache.org/axis/.

[46]The Gene Ontology. 2009. www.geneontology.org/.

[47]GNU Lesser General Public License, GNU Project, Free Software Foundation (FSF). 2009. http://www.gnu.org/licenses/lgpl.html.

[48]PHYLIB Home Page. 2009. http:// evolution.genetics.washington.edu/phylip.html.

[49]AJAX Command Definition. 2009. http://emboss.sourceforge.net/developers/acd/.

[50]Ensembl Genome Browser. 2009. http://www.ensembl.org/index.html.

[51]Homo sapiens-UniGene. 2009. http://www.ncbi.nlm.nih.gov/UniGene/UGOrg.cgi?TAXID=9606.

[52]Haman (Homo sapiens). 2009. http://ensembl.genomics.org.cn:8050/Homo_sapiens/index.html.

[53]Entrez cross-database search. 2009. http://www.ncbi.nlm.nih.gov/sites/gquery.

[54]OMIM Home. 2009. http://www.ncbi.nlm.nih.gov/sites/entrez?db=omim.

[55]MOTIF: Searching Protein and Nucleic Acid Sequences Motifs. 2009. http://motif.genome.jp/.

[56]KEGG: Kyoto Encyclopedia of Gene and Genomes. 2009. http://www.genome.jp/kegg/.

[57]KEGG Pathway Database. 2009. http://www.genome.ad.jp/kegg/pathway.html.

[58]XML Path Language (XPath). 2009. http://www.w3.org/TR/xpath.

[59]Han J, Kraft P, Nan H, Guo Q, Chen C, et al. (2008)., “A Genome-Wide Association Study Identifies Novel Alleles Associated with Hair Color and Skin Pigmentation,” PLoS Genet 4(5): e1000074. doi:10.1371/journal.pgen.1000074.

[60]Damkliang, K., Tandayya, P., Pasomsuba, E., Chantratita, W., and Mahasirimongkol, S. 2009. “Taverna Workflow for Validating Biomart Services.” Proceedings of 2009 International Conference on Computer Design and Applications (ICCDA 2009), Singapore, May 15-17, 2009. pp. 459-463.

[61]Axel M Hillmer, Felix F Brockschmidt, Sandra Hanneken, Sibylle Eigelshoven, Michael Steffens, Antonia Flaquer, Stefan Herms, Tim Becker, Anne-Katrin Kortüm, Dale R Nyholt,

Zhen Zhen Zhao, Grant W Montgomery, Nicholas G Martin, Thomas W Mühleisen, Margrieta A Alblas, Susanne Moebus, Karl-Heinz Jöckel, Martina Bröcker-Preuss, Raimund Erbel, Roman Reinartz, Regina C Betz, Sven Cichon, Peter Propping, Max P Baur, Thomas F Wienker, Roland Kruse and Markus M Nöthen, “Susceptibility variants for male-pattern baldness on chromosome 20p11,” Nature Genetics 40, 1279-1281 (2008). Published online: 12 October 2008 | doi:10.1038/ng.228ใ

[62]ศูนย์วิจัยเภสัชพันธุศาสตร์และสารสนเทศ. 2009. http://www.pharmagtc.net/.

[63]K. Podesta, M. Crane, H.J. Ruskin, "A sequence-focused parallelisation of EMBOSS on a cluster of workstations," International Conference on Computational Science & its Applications (ICCSA '04), Assisi, Italy, May 2004, Publication in Springer Lecture Notes in Computer Science.

[64]Biowulf at the NIH. National Institutes of Health (NIH). 2009. http://biowulf.nih.gov/.

[65]Swarm on the Biowulf Linux Cluster. National Institutes of Health (NIH). 2009. http:// biowulf.nih.gov/apps/swarm.html.

ภาคผนวก ก

บทความวิชาการระดับนานาชาติชิ้นที่ 1

เรื่อง

Taverna Workflow and Supporting Services for

Single Nucleotide Polymorphisms Analysis

งานประชุม

2009 International Conference on Information Management and Engineering

(ICIME 2009), Kuala Lumpur, Malaysia, 3rd - 5th April 2009.

PP. 27-31. ISBN: 978-1-4244-3774-0.

2009 International Conference on Information Management and Engineering

Taverna Workflow and Supporting Services for Single Nucleotide Polymorphisms Analysis

Kasikrit Damkliang[1] and Pichaya Tandayya[2]
Department of Computer Engineering,
Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla Thailand
s5010120123@psu.ac.th[1], pichaya@coe.psu.ac.th[2]

Theerawut Phusantisampan[3] and Unitsa Sangket[4]
The Centre for Genomics and Bioinformatics Research,
Prince of Songkla University,
Hat Yai, Songkhla, Thailand
5110230005@psu.ac.th[3], unitsa.s@psu.ac.th[4]

***Abstract*—This paper describes a Taverna workflow environment with supporting web services for Single Nucleotide Polymorphisms (SNP) analysis of genes in shrimp. The workflow is used to investigate SNPs advantages for gene development. Our first workflow environment did not give satisfying results for the phylogenetic tree processes because the data flows between the web services would frequently time-out. Our solution employs local web service interfaces in a cluster, which Taverna accesses with the aid of Soaplab analysis tools. The cluster enables the modified workflow to rapidly process the large amount of sequenced data required by phylogenetic trees, and is highly reliable.**

Keywords-Taverna, Workflow, Web Service, SNP, Phylogenetics

I. INTRODUCTION

Bioinformatic researchers use to resort to copying and pasting between web pages to combine the results of multiple services or programs. This was labor intensive, error prone, and not scalable, due to standard differences, and research complexity [1, 3]. In response, workflow methodologies were developed to help compose the resources into a single logical system.

This paper presents a workflow environment based on Taverna [2, 4, 5] for Single Nucleotide Polymorphisms (SNP) [6] analysis of shrimp. We focus on the *Amylase* gene because it is a good model for studying the adaptation processes and phylogeny. *Penaeus merguiensis* samples were collected from three of a multigene family regions of The Gulf of Thailand : Trad (TDE, N=3), Songkhla (SKE, N=3), and Surat Thani (SRE, N=3) [7]. Two or three independent clones were sequenced for each specimen.

A previous study [7] reported on the genetic variation of *P.merguiensis* in Thailand using nuclear DNA markers (ACT1, D7MICRO and PvAmy). High intraspecific variation in PvAmy (FST= 0.324) was observed. This raises the question of the possiblity of the PvAmy locus tundergoing strong environmental selection, linked to variations in the ecological factors. Also, does the same phenomenon happen in other areas in Southeast Asia, such as the Malaysian peninsula. To examine this question, we study the polymorphism of the *Amy* gene using primer based on a conserved sequence from Exon7-10. This lets us evaluate the level of genetic variation of *P.merguiensis* from the Gulf of Thailand (Trad, Surat Thani and Songkhla). Moreover, SNPs may become very important in future studies of agriculture, such as growth and resistance to disease infection.

A major issue for a composed workflow is when it cannot finish its processing because data flows between the web services time-out in the phylogenetic tree [8] predicting process at the EBI [9]. This occurs when the processing involves for more than 2,000 bootstrapping replicates. It is possible to execute the predicting process locally but then we need to copy and paste data between the workflow, which is too much time-consuming.

A better solution is to build local web services that talk to the workflow, focusing on the building of the phylogenetic tree of nucleotide and Amino Acid sequences. The web service interfaces for each phylogenetic tree process in the workflow can then be accessed by Taverna. Our local web service interfaces were implemented with Soaplab [10] analysis tools.

Our workflows were implemented with BLAST (Basic Local Alignment and Search Tool) [11], multiple sequence alignment in nucleotide and Amino Acids, prediction of the protein secondary structure, and prediction of the nucleotide and Amino Acid phylogenetic tree [8]. We use Taverna to compose these web services into a workflow with services from XML Central of DDBJ [12], EBI [9], Java local services, local web services, and shim services. The correctness of each service was verified by The Centre for Genomics and Bioinformatics Research at Prince of Songkla University.

II. TAVERNA, WORKFLOW AND SOAPLAB INTRODUCTION

The myGrid e-Science Taverna tool is an open source workflow composer to orchestrate bioinformatic web services and existing applications into workflows, or to help compose local or distributed resources into a single logical system [5, 13].

Taverna can access over 30,000 services in the bioinformatics domain, and will grow to include other fields such as astronomy, chemoinformatics, health informatics in the future

978-0-7695-3595-1/09 $25 © 2009 IEEE
DOI 10.1109/ICIME.2009.38

Taverna can add service endpoints, such as WSDL documents, for improved portability, and can access other types of service, such as, local Java services, BioMoby and BioMart services, and Beanshell scripts [5, 13, 14].

Soaplab is a framework for exposing command-line tools as web services [10]. Soaplab Services containers manage web service jobs, such as Tomcat. It offers a various web services, and provides pointers to the AppLab Server which wraps existing command-line tools. Soaplab is a component of the myGrid project [13].

III. CASE STUDY BACKGROUND

A. Analysis of the Copying and Pasting method

SNP analysis used to rely on copying and pasting between web pages and off-line applications. The analysis processes include sequence analysis with NCBI BLAST [11], multiple alignments with ClustalX [15] and GeneDoc [16], translations of nucleotide sequences to Amino Acid sequences using Protein Sequence Analysis (PSA) [17], the study of evolution relationships by phylogenetic tree with BioEdit [18], and the prediction of secondary structure with PSIPRED [19]. The relationship between these processes is shown as a flowchart in Figure 1.

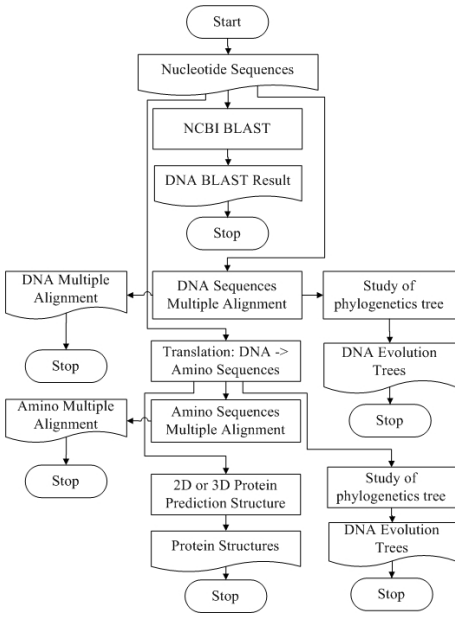


Figure 1. The Flowchart of the Shrimp SNP Analysis

The major web services used for making the workflow are Emboss application services at EBI [9]. Emboss offers many well-respected and broadly known web services.

The BLAST process is invoked by XML Central of DDBJ [20], using configurable databases such as DDBJNEW. BLAST is easy to access via Taverna or a web browser. The workflow also employs the Shim service, a small service that manages the data flow in and out of each service.

The workflow diagram is shown in Figure 2. Table I shows the Soaplab Web Services in the workflow, and the string parameters configure each service [21] are listed in Table II.

In phylogenetics, an evolutionary tree produced with at least 1,000 bootstrapping replicates gives satisfying results because more replicates generate a plausible or correct tree [21]. We plan to utilize 2,000 replicates for precision results.

Figure 2 shows

- The use of existing services by Taverna.
- The prediction of a phylogenetic tree requires the *'fseqboot'* service or a bootstrapped sequences algorithm. *'fseqboot'* reads in a data set, and produces multiple data sets by sampling.
- The workflow input employs the shrimp nucleotide fasta format sequence [22] from a wet lab. There were to about 15-27 sequences; each consisting of 250 base pairs from the Exon7-10. An example of the fasta format sequences is shown below:

```
>SRE26
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGG
AGTTCAGGTACGGCAAGTACCTGGGCGAGGCCTTCCGCGGCAACAACCAGC
TGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGATTGACCGGCATG
ACGCCCTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTGG
AGGAGACATGATCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
>TDE8
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGG
AGTTCAGGTACGGCAAGTACCTGGGCGAGGCCTTCCGCGGCAACAACCAGC
TGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGATTGACCGGCATG
ACGCACTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTG
GAGGAGACATGACCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
>SKE42
GTGGCGAAGCCATATCCAGCGGCGAGTATGTTGGCAACGGTCGTGTGACGG
AGTTCAGGTACGGCAAGTACCTGGGCGAGGCCTTCCGCGGCAACAACCAGC
TGAAATACCTCAACAACTTCGGCGAAGGTTGGGGCATGATTGACCGGCATG
ACGCACTGGTCTTCATTGACAACCACGACAACCAGAGAGGCCATGGTGCTG
GAGGAGACATGATCCTTACTTTCCGTGTCTCTAAGTGGTACAAGGA
```

- The process of predicting a phylogenetic utilizes *'fseqboot'*, *'fdnapars'* or *'fprotpars',* and *'fconsense.'*

The test results are shown in Table III. The workflow did not always produce results when using 2,000 bootstrapping replicates. The data flow between *'fseqboot'* → *'fdnapars'* and *'fseqboot'* → *'fprotpars'* sometimes terminates due to fetching time-out error or inconsistencies in retrieving the results.

B. Design the Solution

We solved the time-out connection problem by building local web services for predicting the phylogenetic tree. The services are hosted on Hanuman, a Rock cluster in the Grid Center at Prince of Songkla University [23].

The Soaplab Analysis Tool was used to implement the web service interfaces. Two types of web services were built, as shown in Figure 3. The application for predicting a phylogenetic tree was wrapped by Soaplab using the nucleotide acid and amino acid parsimony algorithm, I/O was managed using Java web service (JWS) [24].

C. Implementation

PHYLIB 3.6b [25] is an Emboss open source package for predicting a phylogenetic tree. PHYLIB is installed on Hanuman, employing the Rocks Linux Cluster Operating System version 4.3 [26] and Tomcat v.5.0.28 [27] as the web service container.

The key concept is to build an XML services description file for the Soaplab from an ACD [28] file list. An ACD file

must be written for each application program that acts as a web service [10].

TABLE I. EBI SOAPLAB WEB SERVICES USED IN THE WORKFLOW

Service Name	Responsibility
transeq	Translate nucleic acid Sequences to amino acid sequence
garnier	Predict protein secondary structure
fmma_[DNA/Prot]	Multiple alignment program interface to ClustalW program
prettyplot	Display aligned sequences, with coloring and boxing
fseqboot_[DNA/Prot]	Bootstrapped sequences algorithm
fdnapars	DNA parsimony algorithm
fprotpars	Protein parsimony algorithm
fconsense_[DNA/Prot]	Majority-rule and strict consensus tree

TABLE II. STRING PARAMETERS CONFIGURED IN THE WORKFLOW

Parameter Name	Responsibility	Value
Database	Database name used by blast for searching	DDBJNEW
Frame3	Translation using frame 3 [6]	3
osformat_[DNA/Prot]	Output sequences for phylogenetics	phylip
seqtype_[DNA/Prot]	Sequence type for bootstrapping	DNA = d Protein = p
Reps_[DNA/Prot]	Bootstrapping number for replication	Vary on test cases
rootTree_[DNA_Prot]	Produce the phylogenetics root tree	Yes

TABLE III. TESTING CASES AND RESULTS

No of Replicates	Time Used (min.)
100	02.50
250	12.20
500	20.29
1,000	20.40
2,000	Failed (time-out or inconsistency)

The script shown below is an example ACD file for *'fseqboot'*, *'fdnapars'*, *'fprotpars',* and *'fconsense.'* The ACD file configures the web service such as the location of an executable program or input file, the parameter '-auto' is for enabling the application to be non-interactive, the output file system location, or which web service is needed to retrieve the output from the host system

```
string: auto
[ parameter: "Y" comment: "display false" default: "-auto" comment: "defaults" ]
```

The Soaplab acd2xml tool was employed to translate the ACD files list into an XML file, Soaplab deployed the web services using the configuration parameters loaded from the file. The workflow for these local web services was composed in Taverna as shown in Figure 4. The new workflow with local web services is in the black frames.

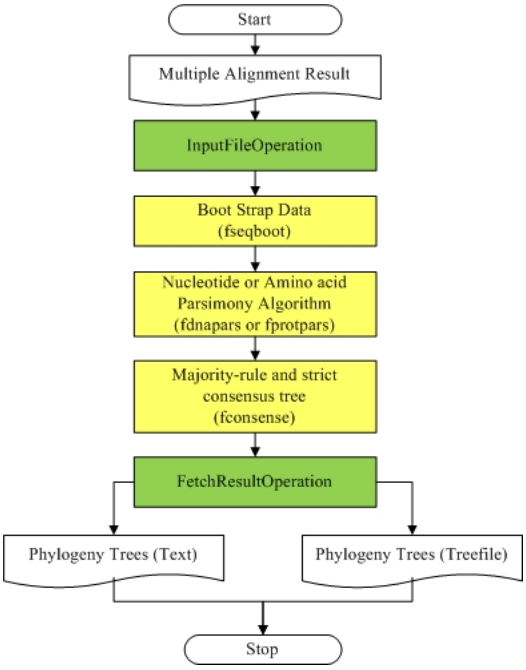


Figure 3. The Flowchart for Predicting the Phylogenetic Tree

IV. EXPERIMENTAL RESULTS

Table IV shows the results of our testing of the Figure 4 workflow, with the number of DNA bootstrapping replicates extended to 50,000. Table V compares manual processing (copying and pasting) versus automatic workflow processing. The first workflow failed due to a time-out error. With 2,000 DNA bootstrapping replicates, manual processing took 15 minutes whereas the automatic processing using the modified workflow took 2.24 minutes. The reduction in processing time is highly significant.

When we increased the number of DNA bootstrapping replicates to 20,000 and then to 50,000; the modified workflow took 15.08 minutes and 36.17 minutes respectively. We did not replicate these tests using manual processing because it is too difficult, or impossible, to manually copy and paste such large amount of data from and to web pages.

Fig. 5 and Fig. 6 show DNA and Amino Acid phylogenetic trees, verified by The Center for Genomics and Bioinformatics at Prince of Songkla University. The evolutionary relationships amongst the three different shrimp sources is quite clear, this knowledge will aid future studies of the selected species.

The DNA phygenetic tree in Figure 5 highlights many genetic relationships. For example, TDE24 and SKE13 are in the same group of species, and have a closer relationship with SRE17 than with SRE16. The Amino Acid phylogenetic tree in Figure 6 shows that SKE50 and SKE42 are in same group of species, and were separated from TDE8.

TABLE IV. TESTING CASES AND RESULTS OF THE NEW WORKFLOW

No of Replicates	Time Used (Min.)
100	02.12
250	02.06
500	02.06
1,000	02.06
2,000	02.24
3,000	03.04
4,000	03.47
5,000	03.52
10,000	08.09
20,000	15.08
30,000	22.02
40,000	29.22
50,000	36.17

TABLE V. TESTING CASES AND RESULTS IN COMPARISON

No of Replicates	Test Case			Running Time (Min.)
	Manual	*Automatic Workflow*		
		No local service	*With local web services*	
2,000	X			15
2,000		X		Failed
2,000			X	2.24
20,000	X			Difficult or impossible to test
20,000			X	15.08
50,000			X	30.17

V. CONCLUSIONS

This paper discusses the design, development and testing of workflows and web services for shrimp SNPs analysis using Taverna and myGrid Technology. Local web services communicating with a BLAST application solve time-out problems and significantly reduce the working time compared to manual operation. The workflow successfully reveals genetic relationships for shrimp. The workflow can also help with analyzing other species, such as mollusc and arthophoda. The user can supply input sequences for the species to the workflow without any modification.

ACKNOWLEDGEMENTS

This research work is a collaboration between the Computer Engineering Department, the PSU Grid Center, and The Centre for Genomics and Bioinformatics Research at Prince of Songkla University. This work is partly supported by the Biogrid Project funded by the Thai National Grid Center and Thailand's Software Industry Promotion Agency. The authors are grateful to Prof. Carole Goble, Prof. Andy Brass and Dr. Katy Wolstencroft at the School of Computer Science at the University of Manchester, and myGrid team members for Taverna training. The authors are also thankful to Dr. Andrew Davison for proof reading the paper.

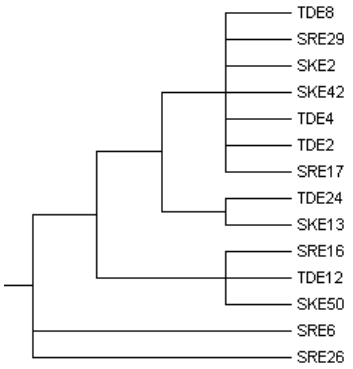


Figure 5. DNA Phygenetic tree

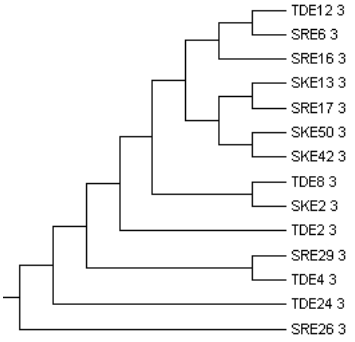


Figure 6. Amino Acid Phylogenetic tree

REFERENCES

[1] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M.,Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A. and Li, P., "Taverna: a tool for the composition and enactment of bioinformatics workflows," BIOINFORMATICS, 2004 Vol. 20, no. 17, pp. 3045–3054, doi:10.1093/bioinformatics/bth361.

[2] Wolstencroft, K.; Oinn, T.; Goble, C.; Ferris, J.; Wroe, C.; Lord, P.; Glover, K.; Stevens, R., "Panoply of utilities in Taverna," First International Conference on e-Science and Grid Computring, 5-8 Dec. 2005.

[3] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Peter Li, P. and Oinn, T., "Taverna: a tool for building and running workflows of services," Nucleic Acids Research, 2006, Vol. 34, Web Server issue W729–W732.

[4] Turi, D., Missier, P., Goble, C., De Roure, D., Oinn, T., "Taverna Workflows: Syntax and Semantics, e-Science and Grid Computing," IEEE International Conference on 10-13 Dec. 2007, pp. 441 – 448.

[5] Taverna project website. 2008. http:// tavern.sourceforge.net.

[6] SNP Home. 2008. http:// www .Ncbi .nlm .nih.gov/sites/entrez?db=snp.

[7] Wanna, W., Rolland, J.L., Bonhomme, F. and Phongdara, A., "Population genetic structure of Penaeus merguiensis in Thailand based on nuclear DNA variation," J. Exp. Mar. Bio, 2004, Ecol. 311, pp. 63-78.

[8] Phylogenetic trees. 2008. http:// cnx .org /content /m11052 /latest/.

[9] EBI Web Services for EMBOSS-4.1 application. 2008. http:// www .ebi .ac .uk /soaplab/emboss4/

[10] Soaplab. 2008. http:// www .ebi .ac .uk /soaplab.

[11] BLAST: Basic Local Alignment and Search Tool document. 2008. http:// www .ncbi .nlm.nih.gov/blast/.

[12] Web API for Bioinformatics. DDBJ. 2008. http:// xml.nig.ac.jp/index.html.

[13] myGrid. 2008. http://www.mygrid.org.uk/.

[14] Biological web services. 2008. http:// www .mygrid .org .uk /wik /myGrid /Biological WebServices.

[15] Using ClustalX. 2008. http:// www .bioinformatics .ubc .ca/ resources /tutorials /using_clustalx.

[16] GeneDoc. 2008. http:// www .nrbsc .org /gfx /genedoc/.

[17] PSA Structure Prediction. 2008. http:// bmerc-www .bu .edu/psa/.

[18] BioEdit Sequence Alignment Editor for Windows 95 /98 /NT /XP. 2008. http:// www .mbio.ncsu.edu/BioEdit/BioEdit.html

[19] The PSIPRED protein structure prediction server. 2008. http:// bioinf .cs .ucl .ac .uk /psipred/.

[20] BLAST. XML Center of DDBJ. 2008. http:// www.xml.nig.ac.jp/wabi/Method?serviceName=Blast&mode=methodList&lang=en

[21] EMBOSS explorer. 2008. http:// pro .genomics .purdue .edu/cgi-bin/emboss.

[22] FASTA format description. 2008. http:// www.ncbi.nlm.nih.gov/blast/fasta.shtml.

[23] Hanuman. Grid Center of Prince of Songkla University website. 2008. http:// hanuman .psu.ac.th/.

[24] JWS. Web Services – Axis. 2008 http:// ws .Apache.org/axis.

[25] PHYLIB Home Page. 2008. http:// evolution . genetics. washington.edu/phylip.html

[26] Rocks Clusters. 2008. http:// www .Rocksclusters.org/.

[27] Apache Tomcat. 2008. http:// tomcat .apache .org/download-55.cgi.

[28] AJAX Command Definition. 2008. http:// emboss.sourceforge.net/developers/acd/.

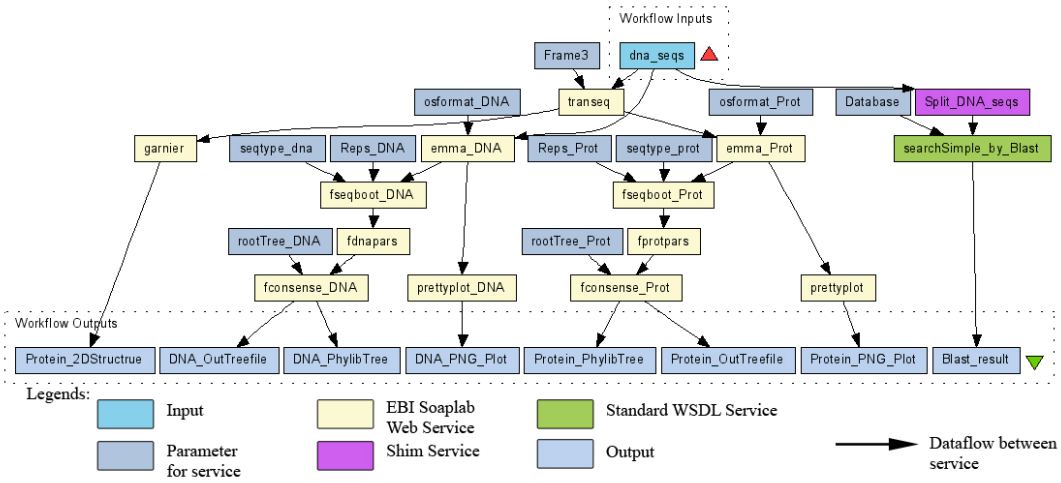
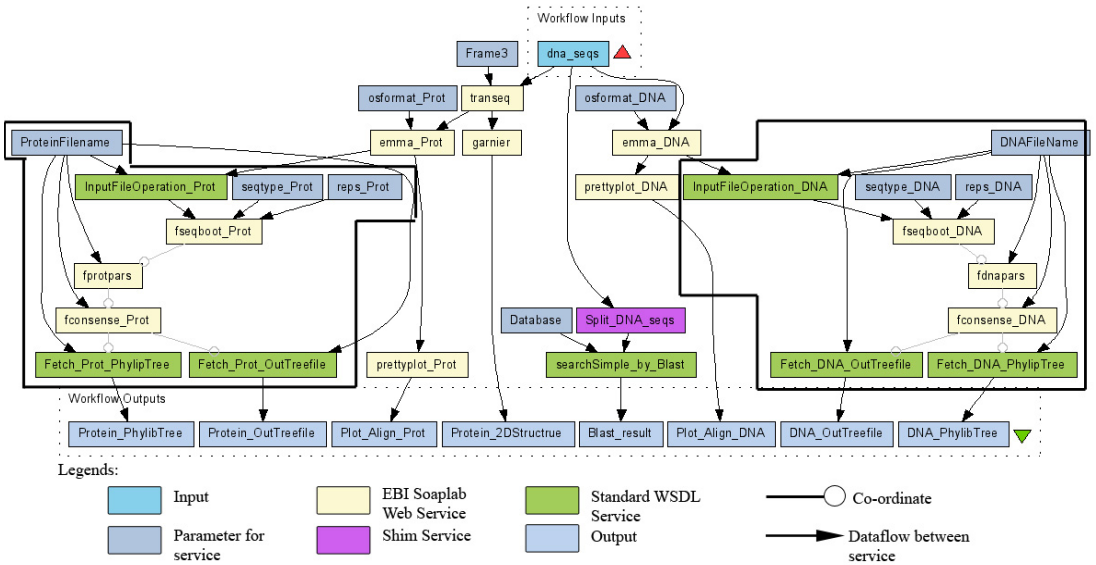


Figure 2. The first workflow of the flowchart



The new workflow uses local web services is in the black frame

ภาคผนวก ข

บทความวิชาการระดับนานาชาติชิ้นที่ 2

เรื่อง

Taverna Workflow for Validating Biomart Services

งานประชุม

2009 International Conference on Computer Design and Applications

(ICCDA 2009), Singapore, 15th - 17th May 2009.

PP. 459-463. ISBN: 978-1-4244-4213-3

2009 International Conference on Signal Processing Systems

Taverna Workflow for Validating BioMart Services

Kasikrit Damkliang[1] and
Pichaya Tandayya[2]
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University
Hat Yai, Songkhla, Thailand
s5010120123@psu.ac.th[1],
pichaya@coe.psu.ac.th[2]

Ekawat Pasomsub[3]
and Wasun Chantratita[4]
Department of Clinical Pathology
Faculty of Medicine Ramathibodi
Hospital, Mahidol University
Bangkok, Thailand
g4737429@student.mahidol.ac.th[3],
rawct@mahidol.ac.th[4]

Surakameth Mahasirimongkol
Center for International Cooperation
Department of Medical Sciences
Ministry of Public Health
Nonthaburi, Thailand
surakameth.m@dmsc.mail.go.th

***Abstract*—This paper present a novel Taverna workflow for validating BioMart Services. It saves a significant amount of time by avoiding incorrect workflows caused by out-of-date services. It also reduces debugging time by finding out-of-date processor query fields (filters) and output fields (attributes). In addition, it provides associated information required for future validation.**

Keywords-workflow; BioMart; Taverna; Web Services; validation

VI. INTRODUCTION

Workflow is a data flow methodology where output of one component forms the input of others [1][6]. If any service in the workflow produces erroneous results, then the entire data flow in that section will produce incorrect results.

To decide whether a workflow gives correct results or not, we normally need to run the workflow and analyse its results. If the workflow is complex, with a lot of data to analyse, then it will take a long time to produce results. The results cannot be guaranteed until they are verified with other research work or data [2]. Therefore, validating the workflow before running it would help avoid unnecessary and non-productive processing, and reduce the working time.

Taverna version 1.7.0 (and later) has a workflow existence plug-in, which can check responsiveness of web service endpoints [3][4]. However, it cannot detect upgrade changes to web services, which produce empty result lists, even though the endpoints are correct.

This problem also impacts the search for processor query fields (filters) and processor output fields (attributes) which may simply be out-of-date [5]. The more complex the workflow, the more time will be lost in running and debugging the workflow.

The test case described here is a complex pharmacogenomic research workflow, requiring BioMart Web services to gather information from databases. The BioMart system is a flexible data warehouse aimed at complex interlinked biological data sets. Taverna's BioMart query integration provides full search and retrieval functionality over these data sources [4][7].

This paper presents a workflow method based on the Taverna environment which validates the BioMart services before they are utilized in the processor query fields and processor output fields. This detects out-of-date services, filters and attributes, greatly reducing debugging times.

VII. BACKGROUND

Many useful bioinformatic databases are accessible via the Internet, but finding the most suitable for a task is difficult since there are a few standards for representing and sharing data. myGrid e-Science provides an analysis method depending on a workflow system which can convert data to comply with web service interfaces and direct the flow of data between resources [1][6][8]. The myGrid e-Science Taverna tool is an open source workflow composer for orchestrating bioinformatic web services and existing applications into workflows. In essence, Taverna helps compose local and distributed resources into a single logical system, by employing Web Service technology [4]

Taverna can access over 30,000 services or processors in the bioinformatics domain, and there are plans to include other fields such as astronomy, chemoinformatics, and health informatics [9].

Taverna can add service endpoints, such as Web Service Definition Language (WSDL) documents [10], for improved portability, and can access other types of resources, such as, local Java services, BioMoby, beanshell scripts, and BioMart services [4][7].

BioMart [7][11] is a widely used query-oriented data management system jointly developed by the Ontario Institute for Cancer Research (OiCR) and the European Bioinformatics Institute (EBI). It is open source and freely distributed without restrictions. The system accepts various types of data and also provides a kind of 'data mining.' Service configuration is achieved through a graphical user interface based on applications using web services, a dedicated API. A BioMart service configuration is saved into a Simple Conceptual Unified Flow Language (Scufl) [4] document in the workflow. BioMart is available to Taverna as a plug-in, providing good ways to compose workflows together.

Taverna version 1.7.0, and later, has a *'Taverna 2 preview'* plug-in for performing workflow health checks. The plug-in examines whether the workflow can be translated from the old style Scufl model into the T2 style, and whether the web service endpoints are responding [4]. It

978-0-7695-3654-5/09 $25.00 © 2009 Crown Copyright
DOI 10.1109/ICSPS.2009.132

cannot detect upgrade changes for BioMart processor filters and attributes, or examine services inside nested workflows. As a consequence, *'Taverna 2 preview'* is somewhat lacking for validating services and reducing debugging time.

The following running and debugging steps are described in order to show how difficult it is for a user to check and debug a workflow by themselves.

When a user knows a workflow has produced incorrect results, he will check the data flow by verifying the input and output of each service and processor, which should highlight the erroneous resource. If not, then the data type and syntax of the resources must be examined. Even if these seem alright, the workflow may still produce incorrect results.

In that case, the problem may actually lie within the processor. Only then will the out-of-date processor filters and processor attributes be found.

Clearly, the main problem is when the BioMart services are upgraded, causing workflows to produce incorrect data flow.

The out-of-date filters or attributes can be found manually by looking for suspicious resources on the configuration screen for the BioMart processor node. If certain filters or attributes are absent then, it means they are out-of-date. However, manual validation can take many hours or days.

VIII. BioMart Workflow Validation

A. Design

A workflow composed in the Taverna environment is saved into a Scufl document, a standard format from myGrid. The Scufl employs an XML (Extensible Markup Language) [12] so documents can be used with the XML Path Language (XPath) [13]. XPath is employed to extract data elements from the Scufl of a BioMart processor.

The BioMart processor is configured as a 'data mining' resource, which consists of a processor, or service name, and a dataset. The dataset is made of information extracted from the configuration databases; for example, the *'Homo sapiens SNPs'* dataset takes data from the dbSNP127, ENSEMBL, TSC1 and HGVbase15 databases [7][8].

The BioMart dataset configuration is presented with the query fields (filters) and output fields (attributes) matching the user's requirements [5]. The BioMart service structure is shown in Figure 1.

A BioMart Service		
A Processor/Service Name		
A Dataset	Query fields (Filters)	Output field (Attributes)

Figure 3. The BioMart service structure in a Taverna workflow

An example Scufl workflow implementation of a BioMart processor is shown below:

```
<s:processor name="hsapiens_gene_GO">
  <biomart:query>
    <biomart:Dataset name="hsapiens_gene_ensembl">
      <biomart:Attribute name="go" />
      <biomart:Attribute name="evidence_code" />
      <biomart:Attribute name="go_description" />
      <biomart:Attribute name="ensembl_gene_id" />
      <biomart:Filter name="ensembl_gene_id" value="" list="true"/>
    </biomart:Dataset>
  </biomart:Query>
</s:processor>
```

When Taverna opens a workflow, the responsiveness of the endpoints will be checked. If any problematic endpoints are found then Taverna will issue a warning message, and not run the workflow [4]. Since Taverna handles the endpoints, our work concentrates on validating the dataset filters and attributes in the processors. A data flow context diagram for the BioMart validating workflow is shown in Figure 2.

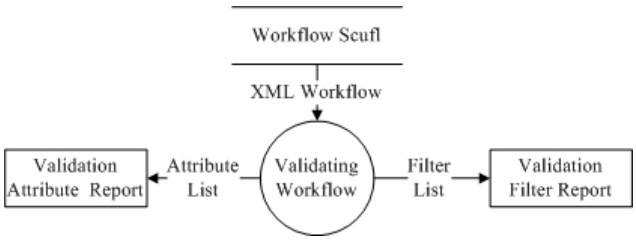


Figure 4. A data flow context diagram for BioMart debugging workflow

The Scufl workflow document(s) fed into the *Validating Workflow* process will produce a filter and attribute validation report.

The processor name used in Figure 3 is extracted from the Scufl document, and is vital for obtaining the dataset name. Subsequently, the Scufl document, processor names and dataset names are fed into a loop of validating processes, as shown in Figure 4.

Fig. 4 describes how out-of-date filters and attributes can be detected by checking their availability with the central BioMart dataset registry on the Internet. If any attribute or filter has been updated, the system collects its description, and details of each processor. This validation report will be utilized to cross-check for BioMart upgrades.

B. Implementation

The nested workflow in Figure 4 acts as an iteration module for the main task. It is utilized to check all the BioMart processor filters and attributes in Scufl documents against the central BioMart registry, located on the BioMart web services server (*http://www.BioMart.org/BioMart /martservice*).

The metadata about service availability is obtained via a GET request using a *'Get web page from URL'* Java local service. A number of metadata queries are available, which can be formulated by appending parameters to the end of the URL [11]. The first parameter is appended using the *?* symbol, and subsequent parameters use the *&* symbol. The format is:

.../martservice?type=<filters|attributes>&virtualschema=de

fault &dataset=<datasetname>

For example, the command for retrieving the attribute registry of the *'hsapiens_gene_ensembl'* BioMart is:

http://www.BioMart.org/BioMart/martservice?type=attributes&virtualschema=default&dataset=hsapiens_gene_ensembl

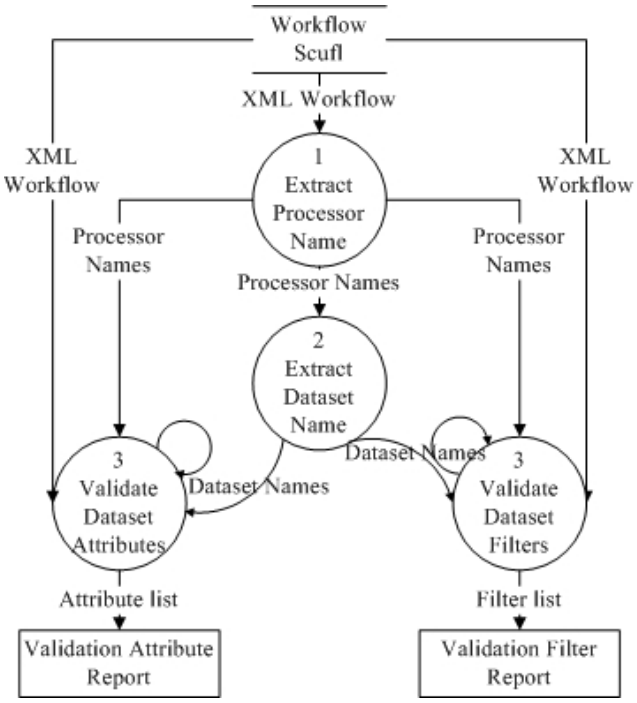


Figure 5. The second level data flow diagram of our validating process

XPath expressions are utilized to extract data elements from the Scufl workflow using the *'Xpath From Text'* Java local service, as shown in Table I. The validating processes are coded in Java using the Beanshell local service [3].

Fig. 5 shows the composed workflow for validating the filters and attributes of large BioMart services. The *'Check_Filter'* and *'Check_Attribute'* rectangular frames are nested looping workflows.

Taverna manages the data flow of output ports using the cross product ('all against all') relationship. In our work, the data flow management for integrating the nested workflows into the upper processes is carried out by the Taverna *'Configure Iterators'* [3] as shown in Figure 6. The processor name and dataset name are used in a dot product ('one against one') relationship, whereas the workflow Scufl employs cross product. The relationship output of the data ports is shown in Table II, showing that a Scufl has many processors, and each processor has only one dataset.

IX. Experimental Results

Our first test utilized an out-of-date pharmacogenomic workflow that no longer produces a correct data flow, and we focused on reducing the validating time and avoiding wasteful executions of workflows that produce no results. The simple testing workflow for Gene Ontology [14] is shown in Figure 7, utilizing a BioMart processor *'hsapiens_gene_ensembl'* labelled as *'hsapiens_gene_GO.'* The service consists of one filter and four attributes. We ran the workflow using two Ensembl IDs. The response arrived in less than two seconds, but was empty.

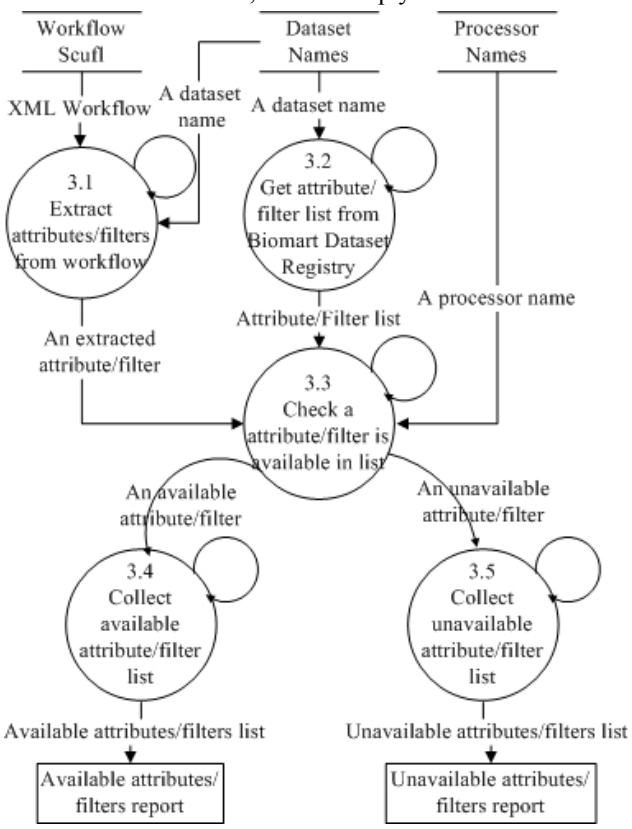


Figure 6. The third level data flow diagram of our validating workflow for dataset filters/attributes

TABLE VI. XPath expression in the workflow

Extracting	Xpath expression
Processor names	//*[(local-name()="Dataset" and (namespace-uri()="http://org.embl.ebi.escience/xScufl-Biomart/0.1alpha")]/ancestor::s:processor/@name
BioMart processor names	//*[(local-name(.)='processor') and (@name=<Processor Name>)] /*[local-name(.)='biomart']/parent::s:processor/@name
BioMart dataset names	/*[(local-name(.)='processor') and (@name=<A BioMart processor name>)] /*[local-name(.)='biomart'] /*[local-name(.)='MartQuery'] /*[local-name(.)='Query'] /*[local-name(.)='Dataset']/@name
BioMart dataset filters or dataset attributes	/*[(local-name(.)='processor') and (@name=<A BioMart processor name>)] /*[local-name(.)='biomart'] /*[local-name(.)='MartQuery'] /*[local-name(.)='Query'] /*[local-name(.)='Dataset'] /*[local-name(.)='Filter' | 'Attribute']/@name

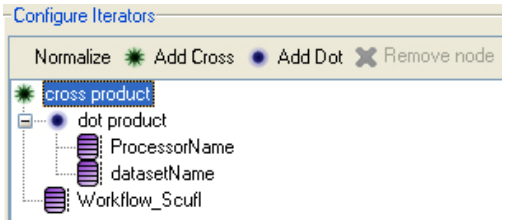


Figure 6. Configure Iterators in Taverna

TABLE VII. THE RESULTS OF CONFIGURE ITERATORS

Cross product	Dot product	Dot product
Workflow_Scufl	1st ProcessorName	1st datasetName
Workflow_Scufl	2nd ProcessorName	2nd datasetName

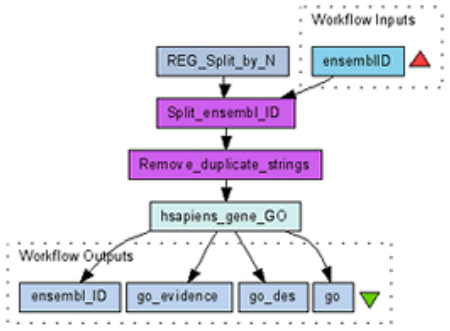


Figure 7. An example of a testing workflow

Our validating workflow helps to identify the out-of-date filters and attributes, as shown below:

Processor Name: hsapiens_gene_GO, Attibute: go is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: evidence_code is out-of-date
Processor Name: hsapiens_gene_GO, Attibute: go_description is out-of-date

The validating workflow also collects the up-to-date filters and attributes for later use:

Processor Name: hsapiens_gene_GO, Filter: ensembl_gene_id is up-to-date Detail: Filter to include genes with supplied list of Ensembl Gene IDs
Processor Name: hsapiens_gene_GO, Attibute: ensembl_gene_id is up-to-date Detail: Ensembl Stable ID of the Gene, feature_page

The description of the *'ensembl_gene_id'* attribute, is the Ensembl Stable ID for the Gene, and is available on the feature page of the BioMart service configuration screen.

The second major test of our validating workflow is a pharmacogenomic workflow, available at http://www.myexperiment.org/workflows/610.

The workflow testing results are shown in Table III. It consists of 52 services, including 17 that link to BioMart services, 70 corresponding data links, and three data querying BioMart services with three filters and 22 attributes.

The workflow inputs are 500 to 1,000 Single Nucleotide Polymorphisms (SNPs) are the smallest genetic statistic P values according to the Genome Wide Association Study (GWAS) [2] of 550,000 SNPs.

It took 14 hours to run the workflow with 1,000 SNP IDs. Every service works successfully except the Gene Ontology part which returns an empty list.

As shown in Table III, it took 26.5 seconds to validate the workflow, and identify the out-of-date filters and attributes at the Gene Ontology finding service. This shows that validating a workflow before executing it can significantly reduce work when there are out-of-date services. Debugging time for finding the cause of incomplete results is also reduced. After the validation, we can configure our workflow so that all the units are up-to-date, as shown at http://www.myexperiment.org/workflows/612.

TABLE VIII. THE PHARMACOGENOMIC WORKFLOW TESTING RESULT

Type	Property	Remark
All Service	52	-
All Data link	152	-
Associated BioMart service	17	3 BioMart query services
Associated BioMart data link	70	-
BioMart Filter	3	3 BioMart query services
BioMart Attribute	22	3 BioMart query services
Test input	1000	SNP IDs
Running time	14.00.50	Hours
Running results	All completed	Except Gene Ontology
Validating time	26.5	Seconds

X. RESULT DISCUSSION AND DEBUGGING SOLUTION

The experimental results show that minor workflow re-design is often required: out-of-date services must be removed and new services added.

Even when there is only one out-of-date filter, the workflow will fail to produce correct data flows because query fields must pass all the filter conditions before retrieving information. The user must fix the offending processor node before the workflow will function.

Even a processor with an out-of-date attribute that is not an input for downstream processes may still cause the processor to produce incorrect data.

Out-of-date filters and attributes due to the upgrades are not shown on the BioMart configuration screen, so the user cannot deselect them. Advanced users may manually edit the Scufl document but this tends to be tedious, confusing and error-prone.

Our advice is to add a new service with the same processor name that occurred in the validation report. This is done by reselecting the service from Taverna's service available panel, and then attaching the processor and its data links to the other processors. When the added processor does not have the necessary filters or attributes, then replacement services in the same domain should be utilized if possible.

XI. CONCLUSIONS

We have developed a Taverna workflow for validating the processor query fields and processor output fields in BioMart services. This validating workflow can help improve workflow processes without requiring the execution of the workflow. It can save significant amounts of computing time when running incorrect workflows, and

reduces the debugging time. It also provides the associated information useful for future validations.

ACKNOWLEDGMENTS

This research work is a collaboration between the Computer Engineering Department, the PSU Grid Center at Prince of Sonkla University, and the Thailand Center of Excellence for Life Sciences (TCELS) at Mahidol University. This work is partly supported by the Biogrid Project funded by the Thai National Grid Center (TNGC), Thailand's Software Industry Promotion Agency (SIPA) and TCELS. The authors are grateful to Prof. Carole Goble, Prof. Andy Brass and Dr. Katy Wolstencroft at the School of Computer Science at the University of Manchester, and myGrid team members for Taverna training. The authors are also thankful for Dr. Andrew Davison for proof reading the paper.

REFERENCES

[1] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Peter Li, P. and Oinn, T., "Taverna: a tool for building and running workflows of services," Nucleic Acids Research, 2006, Vol. 34, Web Server issue W729–W732.

[2] Wellcome Trust Case Control Consortium. "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," Nature, 2007;447, pp. 661-678.

[3] Taverna project website. 2009. http://taverna.sourceforge.net/.

[4] myGrid. 2009. http://www.mygrid.org.uk/.

[5] Rice, P.M.; Bleasby, A.J.; Haider, S.A.; Ison, J.C.; Mcglinchey, S.; Uludag, M., "EMBRACE: Bioinformatics Data and Analysis Tool Services for e-Science," e-Science and Grid Computing, 2006, e-Science '06. Second IEEE International Conference on Dec. 2006, pp. 146 – 146, Digital Object Identifier 10.1109/E-SCIENCE. 2006.261079.

[6] Turi, D., Missier, P., Goble, C., De Roure, D., Oinn, T., "Taverna Workflows: Syntax and Semantics, e-Science and Grid Computing," IEEE International Conference on 10-13 Dec. 2007, pp. 441 – 448.

[7] BioMart. 2009. http://www.BioMart.org/.

[8] Belhajjame, K.; Wolstencroft, K.; Corcho, O.; Oinn, T.; Tanoh, F.; William, A.; Goble, C., "Metadata Management in the Taverna Workflow System," Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on 19-22 May 2008, pp. 651 – 656, Digital Object Identifier 10.1109/CCGRID.2008.17.

[9] Biological web services. 2009. http://www.mygrid.org.uk/wiki /Mygrid/BiologicalWebServices.

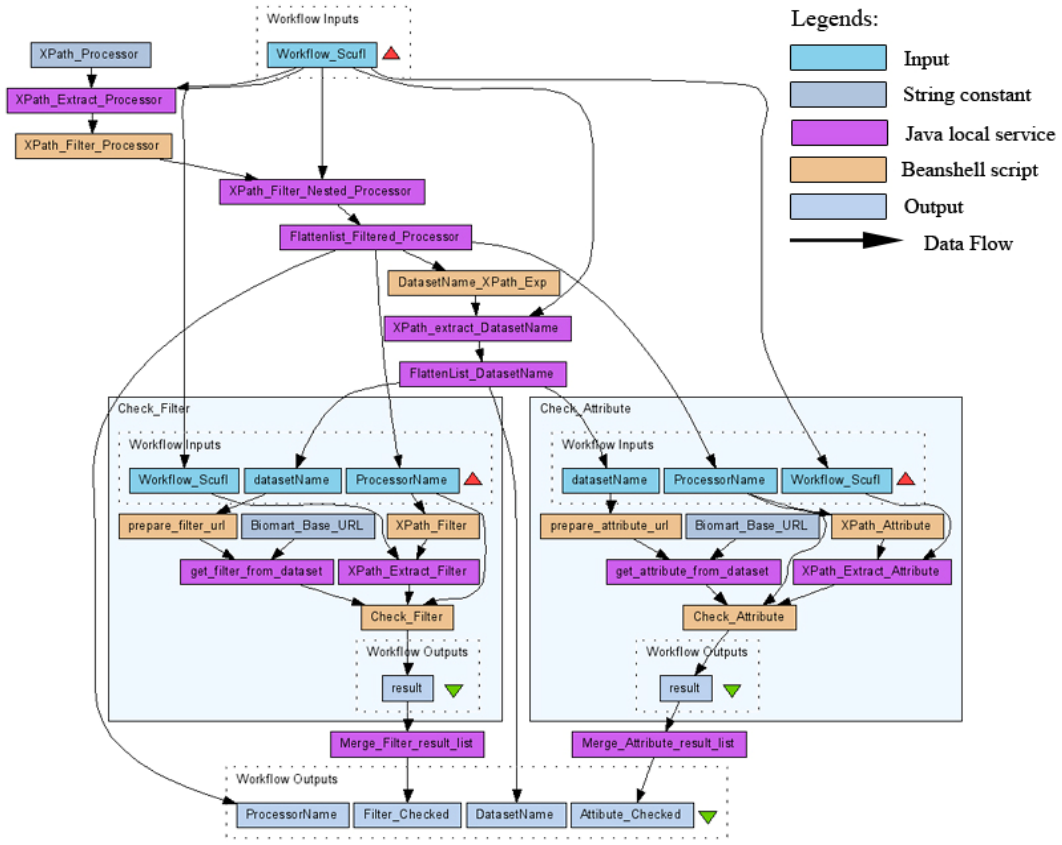
[10] Web Services Description Language (WSDL). 2008. http://www.w3. org/TR/wsdl.

[11] Smedley, D., Haider, S., Ballester, B., Holland, R., London, D., Thorisson, G. and Kasprzyk, A., "BioMart - biological queries made easy," BMC Genomics 2009, 10:22doi:10.1186/1471-2164-10-22.

[12] Extensible Markup Language (XML). 2009. http://www.w3.org /XML/.

[13] XML Path Language (XPath). 2009. http://www.w3.org/TR/xpath.

[14] The Gene Ontology. 2009. http://www.geneontology.org/



The workflow for validating the BioMart filters and BioMart attributes

ภาคผนวก ค

สื่อสาธิตวิธีการทำงานของเวิร์คโฟลว์สำหรับตรวจสอบบริการไบโอมาร์ท

(ดูซีดีที่แนบมากับวิทยานิพนธ์)