



ลอจิกไดอารี่บนเว็บ
Web-Based Logical Diaries

เอมฤดี สุขปิ่น
Amrudee Sukpan

วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

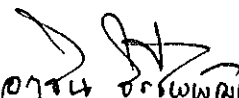
Master of Science Thesis in Computer Science
Prince of Songkla University

2541


เลขที่: ๑๙๗๖.๑.๐๓ ๐๗๔ ๒๕๔๑ ก.๒
Bib Key: ๒๕๑๕๑๘

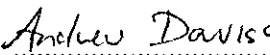
ชื่อวิทยานิพนธ์ ลอจิกไดอาร์บนเว็บ
ผู้เขียน นางสาวเอมฤดี สุขบัน
สาขาวิชา วิทยาการคอมพิวเตอร์

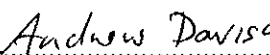
คณะกรรมการที่ปรึกษา



.....
(ดร.อาจิน จิรชีพพัฒนา)

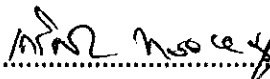
คณะกรรมการสอบ


.....
(ดร.อาจิน จิรชีพพัฒนา)

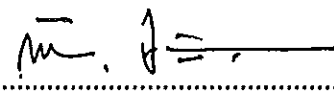

..... กรรมการ
(Dr. Andrew Davison)


..... กรรมการ
(Dr. Andrew Davison)


..... กรรมการ
(ผู้ช่วยศาสตราจารย์ อิศระ ศรีสินทร)


..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.เกริกชัย ทองหนู)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นักวิทยานิพนธ์ ฉบับนี้เป็นส่วน
หนึ่งของการศึกษา ตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์


.....
(รองศาสตราจารย์ ดร.ก้าน จันทร์พรหมมา)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	ลอจิกโคอาร์บินเว็บ
ผู้เขียน	นางสาวเอมฤดี สุขปิ่น
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2541

บทคัดย่อ

การพัฒนาแอปพลิเคชันในปัจจุบันมุ่งพัฒนาเพื่อตอบสนองความต้องการของการติดต่อสื่อสารทางธุรกิจในด้านการจัดการข้อมูลทางอิเล็กทรอนิกส์ เพื่อการพัฒนาสำนักงานที่มีประสิทธิภาพในการทำงานสูง และเพื่อการทำงานร่วมกันกับกลุ่มงานอื่นๆ กลุ่มโคอาร์บีแอปพลิเคชันในปัจจุบันส่วนใหญ่ไม่ได้มีระบบการทำงานอยู่บนเว็บ แต่จะมีการทำงานที่ขึ้นอยู่กับแพลตฟอร์มโคแพลตฟอร์มหนึ่งเท่านั้น โดยข้อมูลของโคอาร์บีที่เก็บจะอยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ (ซึ่งโดยปกติจะใช้ภาษา SQL ในการสอบถามข้อมูล) อย่างไรก็ตามยังมีโคอาร์บีแอปพลิเคชันบางตัวที่พัฒนาขึ้นมาภายหลังที่มีระบบการทำงานอยู่บนเว็บ แต่โคอาร์บีแอปพลิเคชันเหล่านี้ก็ยังคงใช้โครงสร้างข้อมูลและการค้นหาข้อมูลแบบฐานข้อมูลเชิงสัมพันธ์

เป้าหมายของงานวิจัยนี้คือ พัฒนาระบบโคอาร์บินเว็บที่ใช้ภาษาเชิงตรรกะในการทำงาน มีการจัดเก็บข้อมูลของโคอาร์บีไว้ทางด้านเซิร์ฟเวอร์ และทางด้านไคลเอ็นท์สามารถเข้าทำการสอบถามข้อมูลเหล่านั้นได้ในระดับบน ภาษาเชิงตรรกะช่วยให้สามารถเก็บข้อมูลของโคอาร์บีในรูปแบบของฐานข้อมูลเชิงอนุมาณได้ มีการประมวลผลคำถามโดยวิธีการแทนค่าแบบ unification และ การย้อนรอยการทำงานแบบ backtracking ในการค้นหาคำตอบ ด้วยคุณสมบัติเหล่านี้จึงอนุญาตให้สามารถที่จะนำเสนอข้อมูลที่ต้องการค้นหาด้วยวิธีการทำงานอย่างสั้นๆ และทำการค้นหาข้อมูลได้อย่างหลากหลายวิธีที่ไม่สามารถเป็นไปได้ง่ายกับการทำงานกับฐานข้อมูลเชิงสัมพันธ์ แต่ในการทำงานกับภาษาเชิงตรรกะโดยตรงนั้นค่อนข้างจะยากและซับซ้อนสำหรับผู้ที่ไม่มีความรู้ด้านภาษาเชิงตรรกะ ดังนั้นผู้วิจัยจึงได้พัฒนาระบบโคอาร์บีที่มีการทำงานบนเว็บโดยใช้ฟอร์มและกราฟฟิคส์ของเอกสาร HTML เป็นส่วนที่จะติดต่อใช้งานกับผู้ใช้ในระดับบนแทนการติดต่อของผู้ใช้กับการทำงานของภาษาเชิงตรรกะโดยตรง ซึ่งทำให้ผู้ใช้หลายคนอาจจะไม่เชื่อว่าตอนนี้กำลังใช้งานและเขียนคำถามกับโทรลอคแอปพลิเคชันซึ่งทำงานอยู่ทางด้านฝั่งเซิร์ฟเวอร์

Thesis Title	Web-Based Logical Diaries
Author	Miss.Amrudee Sukpan
Major Program	Computer Science
Academic Year	1998

Abstract

Groupware applications have been developed in response to the needs of the business community for electronic data management, office productivity aids, and other team coordination tools. Most diary scheduling applications of this kind are not Web-based, are specific to particular systems and/or platforms, utilise relational database techniques (typically some form of SQL). However, Web-centric tools are becoming available. Even so, these tools still rely on traditional database structuring and search.

The aim of the thesis is to develop a Web-based diary system which uses logic programming for storing the structured server-side data required in a diary, and for specifying high-level client-side queries upon that data. Logic programming has allowed us to develop deductive databases for our diaries, and to formulate queries using unification and backtracking search. This permits us to represent information in a very succinct manner, and to retrieve it in a variety of ways impossible with simpler relational databases. A drawback of logic programming is the difficulty that non-technical users can have with the paradigm. Consequently, we have tried to simplify the interface to our diaries using Web-based graphics and forms; indeed, many users may not realise they are using a Prolog server-side application and writing Prolog queries.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือ และสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอบพระคุณอย่างสูงมา ณ โอกาสนี้คือ

พระเจ้าผู้ทรงประทานสติปัญญาและกำลังให้แก่ผู้วิจัยในการทำวิทยานิพนธ์มาโดยตลอด
ดร.อาจิน จิรชีพพัฒนา อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษา ความรู้ และข้อเสนอแนะทางวิชาการ พร้อมทั้งตรวจทาน และแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัยมาตลอด

Dr.Andrew Davison ผู้เชี่ยวชาญการเขียนโปรแกรมเชิงตรรกในระดับนานาชาติเป็นอาจารย์ปรึกษาร่วม ที่กรุณาให้คำปรึกษา ความรู้ และข้อเสนอแนะทางวิชาการ พร้อมทั้งตรวจทาน และแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัยมาตลอด

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ทุกท่าน ที่สั่งสอนความรู้ทางวิชาการ ซึ่งผู้วิจัยได้นำมาใช้เพื่อเป็นพื้นฐานในการศึกษาและทำวิทยานิพนธ์

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาช่วยตรวจและแก้ไขวิทยานิพนธ์
มหาวิทยาลัยสงขลานครินทร์ซึ่งได้สนับสนุนทุนผลการเรียนดีเด่นสำหรับการศึกษา และการทำวิจัย

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ซึ่งได้สนับสนุนทุนอุดหนุนโครงการพัฒนาซอฟต์แวร์ขนาดเด็กประจำปี 2541

เจ้าหน้าที่ภาควิชาคณิตศาสตร์และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่าน ที่ให้ความช่วยเหลือและอำนวยความสะดวกในการทำวิทยานิพนธ์

เพื่อนๆ น้องๆ คณะบัณฑิตวิทยาลัย ภาควิชาวิทยาการคอมพิวเตอร์ ที่ให้ความช่วยเหลือ คำปรึกษา และเป็นกำลังใจในการทำวิทยานิพนธ์มาโดยตลอด

คุณพ่อ คุณแม่ พี่สาว พี่ชาย และ พี่น้องคริสเตียนทุกท่านที่ได้อธิษฐานเผื่อผู้วิจัย และเป็นกำลังใจในการทำวิทยานิพนธ์มาโดยตลอด

ผู้วิจัยขอให้อองค์พระผู้เป็นเจ้าทรงโปรดอวยพระพรทุกท่านด้วย

เอมฤดี สุขปิ่น

สารบัญ

	หน้า
บทคัดย่อ.....	(3)
Abstract.....	(4)
กิตติกรรมประกาศ.....	(5)
สารบัญ.....	(6)
รายการตาราง.....	(9)
รายการภาพประกอบ.....	(10)
บทที่ 1 บทนำ.....	1
1.1 การตรวจเอกสาร.....	2
1.1.1 โปรแกรมที่ไม่ทำงานบนระบบเครือข่าย.....	3
1.1.2 โปรแกรมที่ทำงานบนระบบเครือข่าย.....	3
1.1.3 โปรแกรมที่ทำงานบนระบบเครือข่ายไฮแมงมุม.....	4
1.2 วัตถุประสงค์.....	4
1.3 ขอบเขตของการดำเนินงาน.....	5
1.4 ขั้นตอนและระยะเวลาการดำเนินงาน.....	5
1.5 สถานที่ และเครื่องมือที่ใช้.....	6
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	7
บทที่ 2 อินเทอร์เน็ต และ เครือข่ายไฮแมงมุม.....	8
2.1 อินเทอร์เน็ต.....	8
2.2 ระบบเครือข่ายไฮแมงมุม.....	8
2.2.1 โครงสร้างสถาปัตยกรรมเครือข่ายไฮแมงมุม.....	9
ขั้นตอนการทำงานของเครือข่ายไฮแมงมุม.....	10
2.2.2 CGI (Common Gateway Interface).....	11
ขั้นตอนการทำงานของเครือข่ายไฮแมงมุมร่วมกับ CGI.....	13
2.2.3 การทำงานร่วมกันระหว่าง CGI และฟอร์ม HTML.....	13
2.2.4 ระบบความปลอดภัยของเว็บเซิร์ฟเวอร์.....	16
ขั้นตอนการทำงานของไฟต์ที่อยู่ภายใต้ไครครทอรี่	
ที่กำหนดผู้มีสิทธิ์เข้าใช้.....	18

2.2.5 จาวาสคริปต์ (JavaScript).....	20
บทที่ 3 โพรลอค.....	23
3.1 ประโยคและลักษณะทั่วไปของการเขียนภาษาโพรลอค.....	24
3.2 BinProlog.....	29
3.3 PiLLow/CIAO Library สำหรับระบบเครือข่ายหรือระบบเครือข่ายไบนารี.....	30
3.3.1 การจัดการฟอร์มด้วย PiLLow.....	30
3.3.2 การเขียนเอกสาร HTML ด้วย PiLLow Library.....	31
บทที่ 4 ลอจิกไดอารีบนเว็บ.....	35
4.1 การทำงานของระบบลอจิกไดอารีบนเว็บ.....	35
4.1.1 ด้านเว็บเซิร์ฟเวอร์.....	35
4.1.1.1 สคริปต์โปรแกรม.....	35
4.1.1.2 ฐานข้อมูลไดอารี.....	37
4.1.1.3 ฟอร์ม HTML 6 ฟอร์ม.....	38
4.1.2 ด้านไคลเอนท์.....	39
4.1.2.1 ผู้ใช้ที่ใช้งานผ่านโปรแกรมบราวเซอร์.....	39
4.1.2.2 โปรแกรมเอเจนต์ลอจิกไดอารีบนเว็บ (Logical Diary Agents on the Web).....	39
4.2 โครงสร้างฐานข้อมูลไดอารี.....	40
4.2.1 ชื่อเท็จจริง diary/3.....	40
4.2.2 กฎ invisible/2 (รูปภาพประกอบ 4.3 ประกอบ).....	43
4.2.3 กฎ reserved/1 (รูปภาพประกอบ 4.3 ประกอบ).....	43
4.3 การสอบถามข้อมูลแบบอนุমানของไดอารี.....	44
4.3.1 การทำงานของโปรแกรมในการสอบถามข้อมูล.....	44
4.3.2 การสร้าง diary/3.....	53
4.3.3 การเปลี่ยนเทอมเงื่อนไขของไดอารีที่ผู้ใช้กำหนดไปเป็นเทอมของระบบ.....	56
4.4 การเปลี่ยนแปลงชื่อเท็จจริง diary/3.....	58
4.4.1 การเพิ่ม diary/3.....	58
4.4.2 การลบ diary/3.....	64
4.4.3 การแก้ไข diary/3.....	66
4.5 การกำหนดกฎ และยกเลิกกฎ reserved/1 และกฎ invisible/2.....	68

4.5.1 การเพิ่มและลบกฎ reserved/1.....	69
การเพิ่มกฎ reserved/1.....	72
การลบกฎ reserved/1.....	72
ตัวอย่างกฎ reserved/1 เพิ่มเติม.....	73
4.5.2 การเพิ่มและลบกฎ invisible/2.....	74
การเพิ่มกฎ invisible/2.....	77
การลบกฎ invisible/2.....	77
ตัวอย่างกฎ invisible/2 เพิ่มเติม.....	78
4.6 การดูไดอารี่รายเดือน.....	78
บทที่ 5 สรุป.....	83
5.1 ผลการวิจัย.....	83
5.2 อุปสรรคและปัญหาในการวิจัย.....	83
5.3 ข้อเสนอแนะ.....	84
บรรณานุกรม.....	85
ภาคผนวก ก ภาพประกอบฟอร์ม HTML ของโปรแกรม.....	87
ภาคผนวก ข การติดตั้งระบบ.....	92
ข.1 ความต้องการของระบบ.....	92
ข.1.1 ด้านเซิร์ฟเวอร์.....	92
ข.1.2 ด้านไคลเอนท์.....	92
ข.2 ขั้นตอนการติดตั้งโปรแกรมลงด้านเซิร์ฟเวอร์.....	92
ประวัติผู้เขียน.....	96

รายการตาราง

ตาราง	หน้า
1.1 แสดงระยะเวลาการดำเนินงานวิจัย.....	6
3.1 แสดงคำสั่งโปรลอคของ Pillow ไลบรารีในการสร้างเอกสาร HTML.....	32
3.2 แสดงคำสั่งโปรลอคของ Pillow ไลบรารีในการสร้างเอกสาร HTML เพิ่มเติม.....	33

รายการภาพประกอบ

ภาพประกอบ	หน้า
2.1 โครงสร้างสถาปัตยกรรมเครือข่ายใยแมงมุม.....	9
2.2 ลักษณะเอกสารไฮเปอร์เท็กซ์.....	11
2.3 แสดงโครงสร้างสถาปัตยกรรมเครือข่ายใยแมงมุมที่ทำงานร่วมกับ CGI.....	12
2.4 แสดงหน้าจอแรกของระบบลอจิก ไดอารี่บนเว็บเพื่อเข้าสู่ระบบ และ ลงทะเบียนเป็นสมาชิก.....	14
2.5 แสดงหน้าจอผลลัพธ์การทำงานของสคริปต์ m_passwd.cgi.....	15
2.6 แสดงไดอะล็อกบ็อกซ์เพื่อสอบถามชื่อและรหัสเมื่อคลิกที่ Member Entry.....	17
2.7 แสดงการทำงานระหว่างบราวเซอร์กับเว็บเซิร์ฟเวอร์ ในการทำ Authentication.....	18
2.8 แสดงแฟ้ม .htaccess.....	19
2.9 แสดงแฟ้ม .htpasswd.....	19
3.1 แสดงโค้ดโปรแกรมครอบครั้วที่เขียนโดยภาษาไพโรลอค.....	23
3.2 แสดงความสัมพันธ์ระหว่างเพรคดิเคด grandparent/2 และ parent/2.....	25
3.3 แสดงขั้นตอนการ backtrack เพื่อหาคำตอบ predecessor(tom,pat).....	27
4.1 แสดงระบบลอจิกไดอารี่บนเว็บ.....	36
4.2 แสดงแฟ้มข้อมูล John.db.....	37
4.3 แสดงแฟ้มข้อมูล Johh.clause.....	38
4.4.ก แสดงหน้าจอการสอบถามนัดหมายกรณีไม่มีเงื่อนไขของคำถาม.....	46
4.4.ข แสดงคำตอบของคำถามในภาพประกอบ 4.4.ก.....	47
4.5.ก แสดงหน้าจอการสอบถามกรณีมีเงื่อนไขของคำถาม.....	48
4.5.ข แสดงคำตอบของคำถามในภาพประกอบ 4.5.ก.....	49
4.6 แสดงหน้าจอการเพิ่มข้อเท็จจริงที่ 1 ของภาพประกอบ 4.1.....	58
4.7 แสดงหน้าจอการเพิ่มข้อเท็จจริงที่ 2 และ 3 ของภาพประกอบ 4.1.....	59
4.8 แสดงหน้าจอการลบนัดหมายทั้งหมดในเดือนสิงหาคม ค.ศ. 1998.....	64
4.9 แสดงหน้าจอการแก้ไขการนัดหมาย.....	66
4.10.ก แสดงหน้าจอการเพิ่มกฎ reserved/1 กรณีไม่มีเงื่อนไข.....	69
4.10.ข แสดงหน้าจอการเพิ่มกฎ reserved/1 กรณีมีเงื่อนไข.....	70
4.11.ก แสดงหน้าจอการเพิ่มกฎ invisible/2 กรณีไม่มีเงื่อนไข.....	74

4.11.ข แสดงหน้าจอการเพิ่มกฎ invisible/2 กรณีมีเงื่อนไข	75
4.12 แสดงการดูไดอารี่ของ John โดย Jack	79
ก.1 ฟอรัมการสอบถามไดอารี่ (query.htm)	87
ก.2 ฟอรัมการเปลี่ยนแปลง diary/3 (change.htm)	88
ก.3 ฟอรัมการเปลี่ยนแปลงกฎของเวลาสงวน reserved/1 (reserved.htm)	89
ก.4 ฟอรัมการเปลี่ยนแปลงกฎการมองเห็น invisible/2 (invisible.htm)	90
ก.5 ฟอรัมแสดงรายการการนัดหมายรายเดือน (view.htm)	91
ข.1 ภาพแสดงเพิ่มข้อมูล srm.conf บางส่วน	93
ข.2 โครงสร้างไครกทอรี่ WBLD ภายใต้ไครกทอรี่ htdocs	94
ข.3 โครงสร้างไครกทอรี่ logical-diary ภายใต้ไครกทอรี่ cgi-bin	95

บทที่ 1

บทนำ

การจดบันทึกนัดหมายลงบนสมุดไดอารี่เพื่อเตือนความจำในเรื่องต่างๆเช่น การนัดหมาย การประชุม และภาระหน้าที่ที่จะต้องทำในแต่ละวัน เป็นสิ่งที่ดีและเป็นประโยชน์อย่างยิ่ง เพราะทำให้เกิดการบริหารเวลาที่ดีและไม่พลาดค่อนัดหมายสำคัญๆ

โปรแกรมคอมพิวเตอร์สำหรับการจัดการไดอารี่ที่มีอยู่ในปัจจุบันแล้วนั้นแบ่งออกเป็น 3 กลุ่มตามลักษณะระบบการทำงานของโปรแกรม กลุ่มที่ 1 โปรแกรมที่ไม่ทำงานบนระบบเครือข่าย กลุ่มที่ 2 โปรแกรมที่ทำงานบนระบบเครือข่าย และ กลุ่มที่ 3 โปรแกรมที่ทำงานบนระบบเครือข่ายแบบนาม

โปรแกรมทั้ง 3 กลุ่มจะช่วยอำนวยความสะดวกในการจัดการไดอารี่อันได้แก่ การบันทึกนัดหมาย การลบนัดหมาย การยกเลิกนัดหมาย โดยแต่ละกลุ่มจะมีความสามารถในการทำงานที่โดดเด่นแตกต่างกัน¹ แต่โปรแกรมที่มีอยู่เหล่านี้ไม่สามารถอำนวยความสะดวกให้กับผู้ใช้งานด้านการสอบถามคำถามที่ซับซ้อน และการกำหนดกฎเกณฑ์ของไดอารี่ในเชิงตรรกได้ ซึ่งปรากฏในบทความ A New Time Scheduling Application of the Web using Logic Programming and Agents² ดังนั้นผู้วิจัยจึงได้ศึกษาและพัฒนาโปรแกรมลอจิกไดอารี่บนเว็บขึ้นมาเพื่อเป็นโปรแกรมต้นแบบในการจัดการไดอารี่ ที่ซ่อนการทำงานของภาษาเชิงตรรกไว้ด้านหลังการทำงานของระบบเครือข่ายแบบนาม (WWW= World Wide Web, เว็บ/WEB) โดยผ่านทาง Common Gateway Interface (CGI)

ภาษาโปรแกรมเป็นภาษาลังงานคอมพิวเตอร์เชิงตรรกที่เหมาะสมสำหรับการประมวลผลจัดเก็บในรูปแบบฐานข้อมูลนามและการตอบคำถามเชิงตรรกที่หลากหลายและซับซ้อนได้ ภาษาโปรแกรมมีคุณสมบัติที่เด่นคือวิธีการแทนค่า (unification หรือ pattern matching), การประมวลผลเชิงตรรก (Logical inference) การย้อนรอยเพื่อค้นหาคำตอบ (Backtracking) จากฐานข้อมูล การประมวลผลกับโครงสร้างแบบลิสท์ การเพิ่มเติมนิยามใหม่เข้าไปในโปรแกรม

แต่การตั้งงานภาษาโปรแกรมโดยตรงเป็นการทำงานที่ค่อนข้างซับซ้อนและต้องมีความรู้ในเรื่องของภาษาโปรแกรม การนำระบบเครือข่ายแบบนามและฟอร์มของเอกสาร HTML ทำงาน

¹ดูรายละเอียดเพิ่มเติมในหัวข้อ 1.1 การตรวจสอบเอกสาร

²Sukpan, A., Santhithitham, D., Jirachiefpattana, A. and Davison, A., 1998, "A New Time Scheduling Application of the web using Logic Programming and Agents", <http://pangha.cs.psu.ac.th/~amrudee/inet/>.

ระดับบนในการติดต่อระหว่างผู้ใช้แทนที่จะติดต่อกับภาษาโทรลกโดยตรงทำให้ผู้ใช้ทำงานได้ง่าย ระบบเครือข่ายใยแมงมุมเป็นบริการบนอินเทอร์เน็ตที่ได้รับความนิยมสูงสุดในขณะนี้และมีผู้ใช้จำนวนมากที่กระจายอยู่ตามที่ต่างๆ ทั่วโลก สามารถแสดงข้อมูลได้สวยงามและหลากหลายไม่เฉพาะเท็กซ์เท่านั้นแต่รวมไปถึง ข้อมูลมัลติมีเดียได้แก่พวก ภาพ เสียง และ วีดีโอด้วย นอกจากนี้ยังมีระบบรักษาความปลอดภัยในการทำงาน

ดังนั้นโปรแกรมลอจิกโคอาร์รี่บนเว็บจึงมีคุณสมบัติสามารถทำงานได้บนทุกแพลตฟอร์มโดยผ่านโปรแกรมบราวเซอร์ในการจัดการกับโคอาร์รี่เพื่อติดต่อนัดหมายระหว่างโคอาร์รี่ โคอาร์รี่อื่นๆ ในระบบ เชื่อมโยงกับเอกสารอื่นบนระบบเครือข่ายใยแมงมุม มีระบบความปลอดภัยของผู้ใช้ สอบถามข้อมูลในเชิงตรรกโดยการกำหนดเงื่อนไขของคำถามตามความต้องการของผู้ใช้ ผู้ใช้สามารถกำหนด กฎขึ้นสำหรับโคอาร์รี่ของตนเองได้ โดยมีอินเตอร์เฟสที่ง่ายต่อการใช้งานผู้ใช้ไม่จำเป็นต้องรู้ภาษาโทรลกอย่างลึกซึ้งในการใช้งาน

การพัฒนาโปรแกรมลอจิกโคอาร์รี่บนเว็บนอกจากจะทำให้เกิดประโยชน์ทางด้านการมีโปรแกรมโคอาร์รี่เชิงตรรกที่มีประสิทธิภาพและง่ายต่อการใช้งานแล้ว ยังทำให้เกิดการผสมผสานการทำงานระหว่างความรู้ทางด้านระบบเครือข่ายใยแมงมุมกับภาษาเชิงตรรก ซึ่งสามารถนำแนวคิดและเทคนิคที่ใช้กันไปพัฒนาแอปพลิเคชันอื่นๆ ในด้านปัญญาประดิษฐ์ และ ระบบชำนาญการบนระบบเครือข่ายใยแมงมุม ที่เป็นประโยชน์ต่อสังคมและประเทศต่อไปในภายภาคหน้า

1.1 การตรวจเอกสาร

โปรแกรมคอมพิวเตอร์ที่ใช้ในการจัดการโคอาร์รี่ในปัจจุบันมีมากมายและหลากหลาย โดยจุดมุ่งหมายของโปรแกรมโคอาร์รี่เหล่านั้น คือช่วยอำนวยความสะดวกให้แก่ผู้ใช้ในการจัดการนัดหมายต่างๆ โปรแกรมโคอาร์รี่สามารถแบ่งเป็น 3 กลุ่มตามลักษณะระบบการทำงานของโปรแกรม โดยมีหัวข้อที่จะพิจารณาความสามารถในแต่ละกลุ่มมี 4 ประเด็นหลักคือ

1. การทำงานทั่วไป
2. การสอบถามข้อมูล
3. ความปลอดภัยของโคอาร์รี่
4. การควบคุมการทำงานโคอาร์รี่โดยผู้ใช้

1.1.1 โปรแกรมที่ไม่ทำงานบนระบบเครือข่าย

หมายเหตุ หมายเลขที่ระบุบนรายชื่อโปรแกรมข้างล่างนี้เพื่อความสะดวกในการอ้างอิงถึงชื่อโปรแกรมซึ่งจะกล่าวซ้ำในภายหลัง

โปรแกรมที่อยู่ในกลุ่มนี้คือ Calendar Wise^[1], OpenCalendar^[2], Calendar Widgets^[3], Calendar Magic^[4], Wysi Calendar^[5], Happy Calendar^[6], Chrono-Logic^[7] และ FF Event Calendar^[8]

โปรแกรมในกลุ่มนี้มีเป็นจำนวนมากตั้งแต่การทำงานบนระบบปฏิบัติการ Dos3.X^[1] (หมายความว่า Calendar* Wise ทำงานบนระบบปฏิบัติการ Dos3.X), Window3.1X^[2,3,4,5] จนกระทั่งถึง Window95^[6,7,8] การทำงานทั่วไปโปรแกรมจะอำนวยความสะดวกในการบันทึก แก้ไข ลบ การทำนัดหมายที่ใช้เวลาหลายๆ วันจนกระทั่งการแสดงผลปฏิทินรายเดือน รายวัน รายปี ง่ายต่อการใช้งาน นอกจากนี้โปรแกรมมีคุณสมบัติการทำงานที่เพิ่มมาของแต่ละโปรแกรม คือ ทำงานได้หลายภาษา^[2] สามารถเชื่อมโยง (Link) ลิงค์กับฐานข้อมูลอื่นๆ ใน Microsoft Visual Basic^[3] ผู้ใช้สามารถกำหนดรูปแบบการแสดงผลบนหน้าจอได้ เช่น รูปภาพ^[3,5] ผู้ใช้สามารถทำการกำหนดเสียงเพื่อเตือนนัดหมายได้^[3,7,8] การกำหนดวัน,เดือน เริ่มต้นในการทำงาน และ ช่วงเวลาที่ใช้งานในแต่ละวัน^[3]

แต่โปรแกรมเหล่านี้ไม่สามารถเชื่อมโยงการทำงานระหว่างหลายๆ ไดอารี่ได้ ไม่มีระบบความปลอดภัยของข้อมูล ผู้ใช้ไม่สามารถสอบถามข้อมูลได้ ผู้ใช้ไม่สามารถกำหนดการทำงานของไดอารี่ให้เป็นไปตามความต้องการได้

1.1.2 โปรแกรมที่ทำงานบนระบบเครือข่าย

โปรแกรมที่อยู่ในกลุ่มนี้คือ AMF Daily Planner & PIM^[9], Calendar+^[10] และ Calendar Quick^[11]

โปรแกรมในกลุ่มนี้นอกจากการทำงานพื้นฐานการจัดการเวลาทั่วไปดังที่ได้กล่าวในหัวข้อที่ 1.1.1 แล้วนั้น ยังมีความสามารถเพิ่มเติมเนื่องจากการมีการทำงานบนระบบเครือข่ายคือสามารถติดต่อกับไดอารี่ของผู้อื่นเพื่อขอข้อมูลในไดอารี่ได้ สามารถเรียกใช้โปรแกรมจดหมายอิเล็กทรอนิกส์ได้ทันที^[9] สามารถหมุนโทรศัพท์ หรือส่งแฟกซ์ได้อัตโนมัติตามเวลาที่กำหนดไว้^[9] (แต่ต้องการ โปรแกรม VB40032.DLL เพิ่มเติมในการทำงาน) มีโปรแกรมผู้ดูแลระบบควบคุมการทำงาน^[10] สามารถค้นหาข้อมูลตามชนิดของนัดหมายที่โปรแกรมกำหนดให้^[11] สามารถเชื่อมโยงไปหาเอกสารอื่นๆ ในเว็บได้^[11] สามารถระบุนัดหมายที่ไม่ต้องการให้แสดงได้^[11]

แต่โปรแกรมเหล่านี้ไม่มีระบบความปลอดภัยของข้อมูล ซึ่งเป็นสิ่งจำเป็นมากเพราะบุคคลอื่นสามารถที่จะดูไดอารี่ของผู้อื่นได้ ไม่สามารถนัดหมายระหว่างไดอารี่ได้ การค้นหาข้อมูล

ไม่ดีเป็นเพียงการค้นหาแบบที่กึ่งธรรมดา บางโปรแกรมต้องการโปรแกรมเพิ่มเติมเพื่อช่วยในการทำงาน และผู้ใช้ไม่สามารถควบคุมการทำงานของไดอารี่ให้เป็นไปตามความต้องการของผู้ใช้ได้ มีผู้ควบคุมระบบเป็นผู้ดูแลและกำหนดการเข้าใช้งานให้ผู้ใช้

1.1.3 โปรแกรมที่ทำงานบนระบบเครือข่ายแบบมูม

โปรแกรมที่อยู่ในกลุ่มนี้คือ Calendar Internet^[12], iCal^[13], WebCal^[14] และ Lotus Organizer Web Calendar for WinNT^[15]

โปรแกรมในกลุ่มนี้จัดเป็นโปรแกรมที่มีประสิทธิภาพดีที่สุดในทั้ง 3 กลุ่มในด้านการใช้งานเพราะมีการทำงานบนระบบเครือข่ายแบบมูม ถึงแม้ว่าตอนนี้โปรแกรมที่มีอยู่จะยังไม่สมบูรณ์เท่ากับโปรแกรมในกลุ่มที่ 1 (โปรแกรมที่ไม่ทำงานบนระบบเครือข่าย) และ กลุ่มที่ 2 (โปรแกรมที่ทำงานบนระบบเครือข่าย) แต่สามารถพัฒนาประสิทธิภาพการทำงานเพิ่มเติมในภายหลังได้ การทำงานของโปรแกรมจะไม่ขึ้นอยู่กับแพลตฟอร์มสามารถเข้าใช้โปรแกรมได้ทุกที่เพราะทำงานโดยผ่านโปรแกรมบราวเซอร์ที่มีการใช้งานอยู่ทั่วไปเช่น เน็ตสเคป (Netscape) ,โมเสอิก (Mosaic), และ อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet explorer) มีระบบความปลอดภัยในการทำงาน สามารถดูไดอารี่และนัดหมายกับไดอารี่ของผู้อื่นได้ สามารถสอบถามแบบที่กึ่งตามชนิดของนัดหมาย^[12,14] และเวลาว่างตามช่วงเวลาที่กำหนดได้^[14]

โปรแกรมในกลุ่มนี้เป็นโปรแกรมที่ทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้กว้างขวางขึ้นและมีระบบความปลอดภัยของข้อมูลเพราะมีการทำงานบนเว็บ แต่โปรแกรมเหล่านี้ไม่สามารถทำงานในด้านการตอบคำถามที่มีความซับซ้อนของผู้ใช้ได้ ผู้ใช้ไม่สามารถกำหนดกฎเกณฑ์การทำงานของไดอารี่ตนเองได้ ในที่นี้ไม่ได้หมายความว่าถึงสภาพแวดล้อมในการทำงานเช่นชนิดของอักษร หรือสี แต่หมายความว่าผู้ใช้สามารถระบุการทำงานของไดอารี่ได้เช่น การกำหนดเวลาที่ไม่ว่าง การไม่อนุญาตให้ดูนัดหมายบางนัดหมาย และเฉพาะบุคคลที่ไม่สามารถดูได้เป็นต้น

1.2 วัตถุประสงค์

1. เพื่อศึกษาและพัฒนาโปรแกรมค้นแบบไดอารี่เชิงตรรกที่ซ่อนการทำงานของภาษาเชิงตรรกไว้ภายใต้ระบบเครือข่ายแบบมูม

2. เพื่อศึกษาและพัฒนาโปรแกรมที่ทำงานบนระบบเครือข่ายแบบไคลเอ็นท์เซิร์ฟเวอร์ (ไคลเอ็นท์/เซิร์ฟเวอร์) ผ่านทาง CGI และการประมวลผลการทำงานทั้งหมดอยู่ทางด้านเซิร์ฟเวอร์ และใช้จาวาสคริปต์ (JavaScript) ในการตรวจสอบการพิมพ์ข้อมูลผิดพลาดของผู้ใช้ทางด้านไคลเอ็นท์

3. เพื่อพัฒนาต้นแบบโปรแกรมไดอารี่ที่อำนวยความสะดวกสำหรับการจัดการไดอารี่ และสามารถกำหนดกฎการทำงานได้ 2 กฎ คือการกำหนดกฎเวลาที่ไม่ว่าง และ การไม่อนุญาตให้ดูนัดหมายของไดอารี่

4. เพื่อศึกษาการทำงานของระบบความปลอดภัยของระบบเครือข่ายเอนกประสงค์และนำมาพัฒนาร่วมกับระบบความปลอดภัยของโปรแกรมไดอารี่

1.3 ขอบเขตของการดำเนินงาน

ในการทำการวิจัยครั้งนี้มีขอบเขตการทำงานดังนี้

1. พัฒนาต้นแบบโปรแกรมไดอารี่ที่มีการทำงานบนระบบเครือข่ายเอนกประสงค์โดยซ่อนการทำงานของภาษาเซิร์ฟเวอร์ไว้เบื้องหลัง

2. พัฒนาต้นแบบโปรแกรมไดอารี่ที่ทำงานโดยภาษาเซิร์ฟเวอร์กับฐานข้อมูลของไดอารี่แบบอนุमान ให้สามารถอำนวยความสะดวกในการจัดการไดอารี่ได้คือ การนัดหมาย การลบ การเปลี่ยนแปลงนัดหมาย และการแสดงไดอารี่รายเดือนการนัดหมายนั้นสามารถกระทำระหว่างหลายไดอารี่ได้ มีการส่งจดหมายอิเล็กทรอนิกส์โดยอัตโนมัติให้แก่ผู้ที่ถูกนัดหมาย โดยการนัดหมาย การแสดงไดอารี่ และการถามคำถามจะเป็นไปตามกฎเกณฑ์ที่ผู้ใช้กำหนด สามารถสอบถามคำถามที่ซับซ้อนเซิร์ฟเวอร์ในฐานข้อมูลแบบอนุमानของไดอารี่ได้ทั้งของตนเองและผู้อื่น

3. พัฒนาต้นแบบโปรแกรมเพื่อการกำหนดกฎการทำงานของไดอารี่โดยผู้ใช้ 2 กฎคือ กฎการมองเห็นข้อมูล และ กฎการจองเวลาสงวนของผู้ใช้ลงในฐานข้อมูลแบบอนุमान และกฎ 2 กฎนี้จะถูกนำมาประมวลผลพร้อมกับการประมวลผลฐานข้อมูลไดอารี่

4. ใช้ระบบความปลอดภัยของเครือข่ายเอนกประสงค์ทำงานร่วมกับระบบความปลอดภัยของโปรแกรมไดอารี่

5. พัฒนาโปรแกรมต้นแบบลจิกไดอารี่บนเว็บที่สามารถทำงานร่วมกับโปรแกรมเอเจนต์ลจิกไดอารี่บนเว็บ (Logical Diary Agents on the Web) ซึ่งพัฒนาขึ้นโดย นางสาวเดือนเพ็ญ สันธิศิริธรรมได้

1.4 ขั้นตอนและระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน

1. ตำรวจรวบรวมข้อมูล ศึกษาภาษาเซิร์ฟเวอร์, การทำงานและความปลอดภัยของระบบเครือข่ายเอนกประสงค์, CGI, โปรโตคอล HTTP, ภาษา HTML และ Javascript

2. เลือกเครื่องมือและศึกษาการใช้งาน BinProlog และ PiLLow Library

3. ออกแบบ กำหนดขอบเขตและการทำงานของโปรแกรม
4. ศึกษาและออกแบบฐานข้อมูลโคอาร์ดี
5. พัฒนาโปรแกรมลอจิกโคอาร์ดีสำหรับผู้ใช้ 1 คน โดยใช้คอมพิวเตอร์ BinProlog ในการพัฒนา
6. นำโปรแกรมที่พัฒนาในข้อ 5 มาทำงานบนเว็บ ผ่าน CGI ร่วมกับ PiLLow Library ในการรับแสดงอินพุตและผลลัพธ์ตามลำดับ โดยนำโปรแกรมในข้อ 5 ปรับปรุงร่วมกับ PiLLow Library ทำให้เป็น CGI Script และสร้างฟอร์ม HTML เพื่อใช้ในการติดต่อระหว่างผู้ใช้กับโปรแกรม CGI Script
7. ขยายโปรแกรมในข้อ 6 ให้สามารถทำงานระหว่างหลายๆ โคอาร์ดีได้
8. ทดสอบโปรแกรม และ จัดทำเอกสารประกอบโปรแกรมและงานวิจัย

ระยะเวลาการดำเนินงาน

ใช้เวลาดำเนินงานประมาณ 1 ปี 1 เดือน โดยเริ่มตั้งแต่เดือน มิถุนายน พ.ศ. 2540 ถึงเดือน กรกฎาคม พ.ศ. 2541 ดังรายละเอียดตามตาราง 1.1

ตาราง 1.1 แสดงระยะเวลาการดำเนินงานวิจัย

ขั้นตอน	มิ.ย	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย	พ.ค.	มิ.ย	ก.ค
1	←→													
2		←→												
3			←→											
4			←→											
5				←→										
6					←→									
7											←→			
8				←→										

1.5 สถานที่ และเครื่องมือที่ใช้

สถานที่

ห้องปฏิบัติการคอมพิวเตอร์ M.105 ภาควิชาคณิตศาสตร์ มหาวิทยาลัยสงขลานครินทร์
วิทยาเขตหาดใหญ่

เครื่องมือที่ใช้

1. เครื่องคอมพิวเตอร์ SUN SPARC station Model 110 MHz หน่วยความจำ 32 MB ฮาร์ดดิสก์ 1.05 GB ระบบปฏิบัติการยูนิกซ์ Solaris 2.5.1 เพื่อเป็นเซิร์ฟเวอร์ที่ต่อกับระบบเครือข่าย และมีเซิร์ฟเวอร์ HTTP ทำงานอยู่ โปรแกรมเซิร์ฟเวอร์ที่ใช้งานคือ Apache สามารถดาวน์โหลดมาใช้งานได้จาก <http://www.apache.org> ชื่อเครื่องเซิร์ฟเวอร์คือ pangba.psu.ac.th หมายเลขแอดเดรสคือ 203.154.171.9

2. เครื่องไมโครคอมพิวเตอร์ แรม 16 เมกะไบต์ ฮาร์ดดิสก์ 1.9 กิกะไบต์ ซีพียูเพนเทียม 5/75 สำหรับเป็นเครื่องไคลเอ็นท์

3. ตัวแปลภาษาไพธอน BinProlog 5.75

4. Pillow Library เวอร์ชัน 1.0 ไบรารีของภาษาเชิงตรรกสำหรับทำงานร่วมกับ CGI และการสร้างผลลัพธ์ในรูปแบบเอกสาร HTML

5. โปรแกรมสร้างเอกสาร HTML เช่น FrontPage98

6. โปรแกรม HTTP เซิร์ฟเวอร์

7. โปรแกรมบราวเซอร์ Netscape Communicator และ Internet Explorer 4.0

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โปรแกรมต้นแบบลอจิกไดอารีบนเว็บที่มีการทำงานบนระบบเครือข่ายใยแมงมุม และซ่อนภาษาเชิงตรรกไว้เบื้องหลังการทำงานจากผู้ใช้ภายใต้ระบบเครือข่ายใยแมงมุม ที่สามารถตอบคำถามที่ซับซ้อนและผู้ใช้สามารถกำหนดกฎในการทำงานสำหรับการแสดงข้อมูลและการกำหนดเวลาสงวนได้ ซึ่งสามารถนำไปใช้งานได้จริง และสามารถนำโปรแกรมนี้ไปพัฒนาต่อไป เพื่อให้มี ประสิทธิภาพ ในการทำงานมากยิ่งขึ้น

2. นำโปรแกรมที่พัฒนาขึ้นไปทำงานร่วมกับโปรแกรม Logical Diary Agents on Web ซึ่งพัฒนาขึ้นโดย นางสาวเดือนเพ็ญ สันจิติธรรมได้

3. สามารถใช้เทคนิคและวิธีการในการพัฒนาโปรแกรมนี้ไปช่วยในการพัฒนาโปรแกรมทางด้านระบบชำนาญการบนระบบเครือข่ายใยแมงมุมอื่นๆ ได้

4. ได้เทคนิคและวิธีการซ่อนการทำงานของภาษาเชิงตรรกไว้เบื้องหลังการทำงานของระบบเครือข่ายใยแมงมุมโดยใช้ CGI

5. ได้ศึกษาและใช้ระบบความปลอดภัยของเครือข่ายใยแมงมุมกับโปรแกรมไดอารีที่พัฒนาขึ้น

บทที่ 2

อินเทอร์เน็ต และ เครือข่ายใยแมงมุม

ยุคนี้กล่าวได้ว่าเป็นยุคแห่งข้อมูลข่าวสาร การสื่อสารโดยผ่านระบบเครือข่ายคอมพิวเตอร์จึงเป็นสิ่งที่สำคัญ เพราะเครือข่ายคอมพิวเตอร์ช่วยให้การแลกเปลี่ยนข้อมูลข่าวสาร สะดวก รวดเร็ว และค่าใช้จ่ายที่ถูกกว่าการติดต่อสื่อสารแบบอื่นๆ ระบบเครือข่ายคอมพิวเตอร์ที่กำลังมีบทบาทที่สำคัญครอบคลุมการทำงานทั่วโลก และมีการขยายตัวของเครือข่ายสูงที่สุดคือ อินเทอร์เน็ต

2.1 อินเทอร์เน็ต

อินเทอร์เน็ต คือ เครือข่ายคอมพิวเตอร์ และเครือข่ายของเครือข่ายที่เชื่อมโยงติดต่อกัน ภายใต้โปรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) มีต้นกำเนิดมาจากการพัฒนาการของเครือข่ายในยุคสงครามเย็นระหว่างกลุ่มประเทศในค่ายคอมมิวนิสต์กับค่ายเสรีประชาธิปไตย สหรัฐอเมริกาซึ่งเป็นประเทศผู้นำในค่ายเสรีประชาธิปไตยได้วางโครงการพัฒนาเครือข่ายคอมพิวเตอร์เพื่อการทหารและต่อมาได้ขยายตัวออกไปทั่วโลก จนกระทั่งพัฒนามาเป็นอินเทอร์เน็ตที่เป็นแหล่งรวบรวมข้อมูลแหล่งใหญ่ที่สุดของโลกที่รวมทั้งบริการและเครื่องมือสืบค้นข้อมูลหลากหลายประเภท สำหรับการศึกษ ธุรกิจ บันเทิง และองค์กรทั้งขนาดเล็กและใหญ่ ปัจจุบันมีผู้ใช้ประมาณ 25 ล้านคน มีโฮสต์ที่เชื่อมต่ออยู่มากกว่า 2.2 ล้านเครื่อง

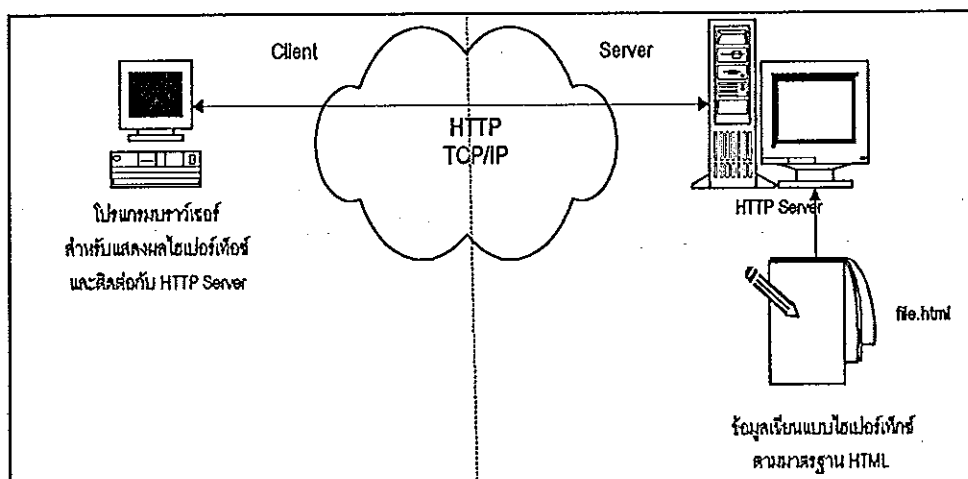
บริการบนอินเทอร์เน็ตในปัจจุบันได้แก่ จดหมายอิเล็กทรอนิกส์ (e-mail), การใช้คอมพิวเตอร์บนเครื่องอื่น (telnet), การถ่ายแฟ้มข้อมูล (ftp), โกเฟอร์ (gopher), การสนทนาผ่านเครือข่าย (talk) และ ระบบเครือข่ายใยแมงมุม (WWW = World Wide Web) หรือที่รู้จักกันดีในชื่อของ เว็บ (WEB)

2.2 ระบบเครือข่ายใยแมงมุม

ระบบเครือข่ายใยแมงมุมเป็นบริการข้อมูลบนอินเทอร์เน็ตที่ได้รับความนิยมสูงสุด จุดเด่นของเครือข่ายใยแมงมุมที่มีเหนือบริการอื่นๆ บนอินเทอร์เน็ตได้แก่ ความง่ายในการใช้งานและรูปแบบการแสดงผลแบบไฮเปอร์เท็กซ์ ข้อมูลในเครือข่ายใยแมงมุมมีทั้งข้อความปกติ ข้อมูลแบบมัลติมีเดีย ที่ประกอบด้วยเสียง ภาพนิ่ง และภาพเคลื่อนไหว

ไฮเปอร์เท็กซ์เป็นแนวคิดการนำเสนอเอกสารและข้อมูล ที่เกิดขึ้นก่อนระบบเครือข่าย โยแมงมูมจนกระทั่ง ทิม เบอร์เนอร์ส-ลี (Tim Berners-Lee) และเพื่อนร่วมงานที่ CERN กรุงเจนีวา ประเทศสวิตเซอร์แลนด์ นำมาพัฒนาและสร้างโปรแกรมอ่านเอกสารมาแสดงบนจอภาพซึ่งเรียกว่า “โปรแกรมบราวเซอร์” (Browser Program) เมื่อโปรแกรมบราวเซอร์เริ่มแพร่หลายและเป็นที่ยอมรับ ในปีพ.ศ. 2537 มาร์ค แอนเดรเสน นักศึกษามหาวิทยาลัยฮิลลินอยส์ที่ เออร์บานา แซมเปญู และอีริก บีน่า ซึ่งเป็นโปรแกรมเมอร์ที่ NCSA ได้พัฒนาโปรแกรมบราวเซอร์ โมเสก (Mosaic) ขึ้น หลังจากนั้นก็มีโปรแกรมบราวเซอร์เกิดขึ้นมาอีกหลายตัว ที่นิยมใช้ในปัจจุบัน ได้แก่โปรแกรม เน็ตสเคป (Netscape) และ อินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet explorer)

2.2.1 โครงสร้างสถาปัตยกรรมเครือข่ายโยแมงมูม



ภาพประกอบ 2.1 โครงสร้างสถาปัตยกรรมเครือข่ายโยแมงมูม

เครือข่ายโยแมงมูมประกอบด้วยการทำงาน 2 ด้านคือด้านไคลเอ็นท์ และด้านเซิร์ฟเวอร์ โปรแกรมบราวเซอร์เป็นโปรแกรมที่อยู่ทางด้านไคลเอ็นท์ ผู้ใช้ใช้สำหรับติดต่อกับ HTTP เซิร์ฟเวอร์ เพื่อร้องขอเพิ่มข้อมูลต่างๆซึ่งส่วนใหญ่จะเป็นเอกสารไฮเปอร์เท็กซ์จากทางด้านเซิร์ฟเวอร์ที่เขียนขึ้นโดยภาษา HTML (HyperText Mark Language) ซึ่งเป็นภาษามาตรฐานในการเขียนข้อมูลแบบไฮเปอร์เท็กซ์ โดยใช้โปรโตคอล HTTP (HyperText Transfer Protocol) ที่มีทำงานอยู่บนโปรโตคอล TCP/IP ติดต่อกันระหว่างไคลเอ็นท์และเซิร์ฟเวอร์ ด้านเซิร์ฟเวอร์จะเป็นผู้ให้บริการคือส่งเพิ่มข้อมูลให้แก่ไคลเอ็นท์ตามที่ร้องขอมา

โปรแกรมบราวเซอร์สามารถเชื่อมโยงและติดต่อกับแหล่งข้อมูลได้หลายรูปแบบหรือหลายโปรโตคอล เช่น โปรโตคอลรับส่งข้อมูลแบบไฮเปอร์เท็กซ์เรียกว่า HTTP (Hypertext Transfer Protocol), โปรโตคอล FTP (File Transfer Protocol) หรือการค้นข้อมูลด้วยโกเฟอร์ (gopher) เป็นต้น

เพื่อให้การเรียกใช้บริการเหล่านี้เป็นไปอย่างมีแบบแผน จึงมีการกำหนดรูปแบบมาตรฐานการเขียนขึ้นเรียกว่า URL¹ (Uniform Resource Locator) เพื่อใช้บอกโปรแกรมบราวเซอร์ว่าต้องการขอใช้บริการใดจากศูนย์บริการแห่งใด การกำหนด URL มีมาตรฐานตามรูปแบบดังนี้

Protocol://Host[:port]/Path/Filename

โดย

Protocol : ชนิดของโปรโตคอลที่ขอบริการเช่น http, gopher, news และ ftp
 Host : ชื่อศูนย์บริการหรือเลขที่อยู่ของโฮสต์คอมพิวเตอร์ในอินเทอร์เน็ต
 port : หมายเลขพอร์ตของโฮสต์ที่ใช้ในการทำการติดต่อ (มีหรือไม่มีก็ได้)
 Path : เส้นทางของแฟ้มข้อมูล
 Filename : ชื่อแฟ้มข้อมูล

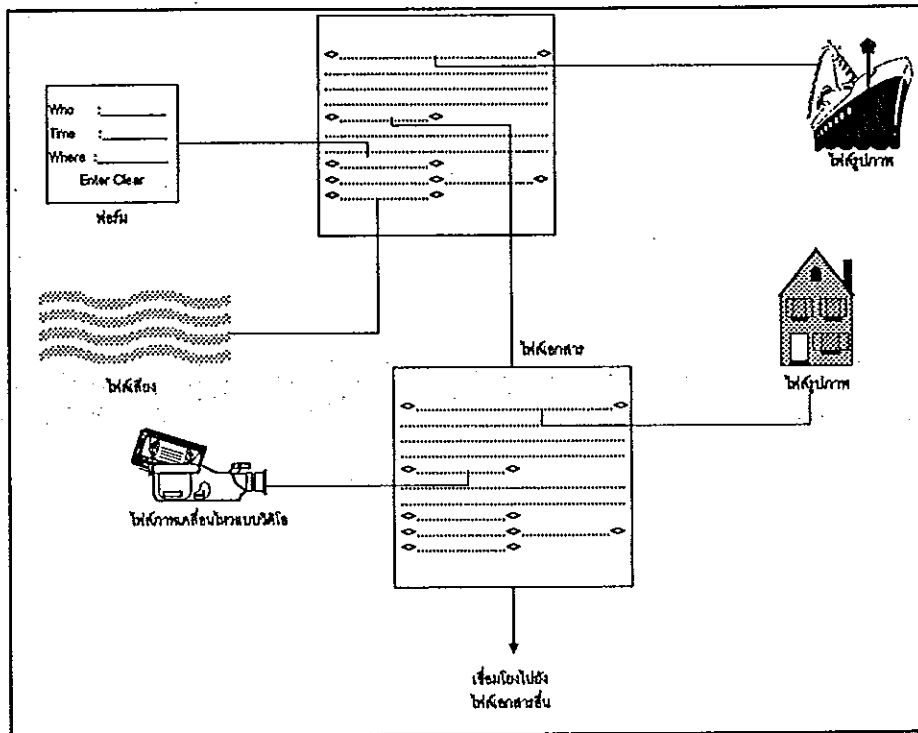
ตัวอย่าง ftp://www.games.com/simc.exe

ขั้นตอนการทำงานของเครือข่ายแมงมุม

สมมติฐานว่าผู้ใช้ระบุ URL ให้แก่โปรแกรมบราวเซอร์เรียบร้อยแล้ว

1. โปรแกรมบราวเซอร์ถอดรหัสส่วนแรกของ URL แล้วติดต่อกับเซิร์ฟเวอร์
2. โปรแกรมบราวเซอร์ส่งส่วนของ URL ที่เหลือไปให้ เซิร์ฟเวอร์
3. เซิร์ฟเวอร์แปลง URL ให้เป็นเส้นทางของแฟ้มข้อมูลในระบบ และ ชื่อแฟ้มเอกสาร
4. เซิร์ฟเวอร์ส่งแฟ้มข้อมูลที่ต้องการไปให้โปรแกรมบราวเซอร์
5. เซิร์ฟเวอร์ยกเลิกการติดต่อกับโปรแกรมบราวเซอร์
6. โปรแกรมบราวเซอร์แสดงแฟ้มข้อมูลที่ได้รับมาบนหน้าจอ

¹ มาจากคำว่า Universal Resource Identifier (URI)



ภาพประกอบ 2.2 ลักษณะเอกสาร ไฮเปอร์เท็กซ์

เอกสารไฮเปอร์เท็กซ์เป็นเอกสารที่มีลักษณะเป็นแฟ้มเอกสารข้อความ โดยข้อความสามารถเชื่อมโยงกับข้อความ แฟ้มเอกสาร ข้อมูลมัลติมีเดีย (แฟ้มนามสกุล gif, tif, wav, และ avi เป็นต้น) และฟอร์มของ HTML เพื่อทำงานกับ CGI Script โครงสร้างของเอกสารไฮเปอร์เท็กซ์ประกอบด้วย ข้อความของเอกสาร (Texts) และ รหัสกำหนดลักษณะการแสดงผลของข้อความเอกสาร (Tags)¹

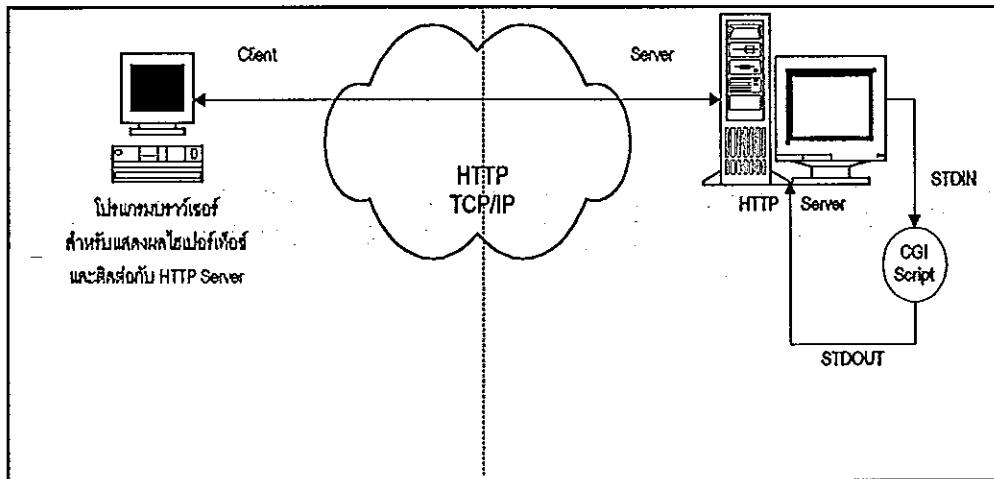
2.2.2 CGI (Common Gateway Interface)

การทำงานของเครือข่ายอินเทอร์เน็ตที่กล่าวในหัวข้อ 2.2.1 นั้นเป็นการทำงานกับแฟ้มเอกสารซึ่งไม่มีการเปลี่ยนแปลงใดๆเมื่อคลิกเอ็นท์ร้องขอ ทำให้ไม่สามารถใช้กับการทำงานของโปรแกรมที่มีข้อมูลแต่ละครั้งในการแสดงผลการทำงานที่ไม่เหมือนกันได้ และไม่สามารถจะรับอินพุตจากผู้ใช้หรือได้ตอบการทำงานกับผู้ใช้ได้

การทำงานของ CGI ทำให้เพิ่มประสิทธิภาพและขยายขอบเขตการใช้งานที่กว้างขึ้นของเครือข่ายอินเทอร์เน็ต ทำให้สามารถสร้างเอกสารที่เป็นแบบไดนามิกและสั่งประมวลผล

¹ ใ้ดค้วอห่งงเอกสารไฮเปอร์เท็กซ์เพิ่มเติมได้ที่หน้า 16

โปรแกรมที่อยู่ทางด้านฝั่งเซิร์ฟเวอร์ได้ ซึ่งผลลัพธ์ของการทำงานในแต่ละครั้งไม่เหมือนกันขึ้นอยู่กับค่าของอินพุตที่ได้รับเข้ามา



ภาพประกอบ 2.3 แสดงโครงสร้างสถาปัตยกรรมเครือข่ายแบบมูมที่ทำงานร่วมกับ CGI

CGI Script คือโปรแกรมที่เขียนขึ้นเพื่อทำงานเฉพาะอย่างอยู่ทางด้านเซิร์ฟเวอร์ การสั่งงานเพื่อรันโปรแกรมจะทำโดย HTTP เซิร์ฟเวอร์ โดย CGI Script จะอยู่ในไคลเรกทอรีเฉพาะหรือมีนามสกุลเฉพาะเท่านั้นเพื่อให้เซิร์ฟเวอร์ทราบว่าเป็นสคริปต์โปรแกรมไม่ใช่เป็นแฟ้มข้อมูลทั่วไป ในปัจจุบันนี้ภาษาที่ใช้เขียน CGI สคริปต์ ได้แก่ ซี (C), วิวอลเบสิก (Visual Basic), เพิลล์ (Perl) และ บินโพรลอค (BinProlog) สำหรับในวิทยาลัยพนธ์ฉบับนี้ใช้ BinProlog ซึ่งเป็นโปรแกรมเชิงครรคในการเขียนสคริปต์ในการทำงานทั้งหมด แต่ในส่วนการเข้ารหัสลับของระบบความปลอดภัยของระบบเครือข่ายนั้นจะต้องใช้ภาษาซีช่วยเพราะต้องใช้ฟังก์ชันในภาษาซีคือฟังก์ชัน `crypt()`¹ เข้ารหัสของผู้ใช้งานระบบเครือข่ายแบบมูมเท่านั้น ซึ่งไม่สามารถจะใช้ BinProlog เขียนได้

อินพุตของสคริปต์รับมาจาก CGI ซึ่งเป็น STDIN และผลลัพธ์ของการทำงานก็จะส่งกลับออกไปทาง CGI ซึ่งเป็น STDOUT เช่นกัน โดยส่วนใหญ่สคริปต์จะส่งผลลัพธ์ออกไปในรูปเอกสาร HTML เพราะมีรูปแบบการแสดงผลเอกสารได้ดีบนโปรแกรมบราวเซอร์กว่าการแสดงผลแบบเท็กซ์ธรรมดา

¹ ดูเห็นข้อมูลภาพประกอบ 2.9 รหัสของยูนิโคดเข้ารหัสโดยฟังก์ชัน `crypt()`

ขั้นตอนการทำงานของเครือข่ายแม่ข่ายร่วมกับ CGI

สมมติฐานผู้ใช้ระบุ URL ของสคริปต์ให้แก่โปรแกรมบราวเซอร์เรียบร้อยแล้ว

1. โปรแกรมบราวเซอร์ถอดรหัสส่วนแรกของ URL แล้วติดต่อกับ เซิร์ฟเวอร์
2. โปรแกรมบราวเซอร์ส่งส่วนของ URL ที่เหลือไปที่ เซิร์ฟเวอร์
3. เซิร์ฟเวอร์ แปลง URL ให้เป็นเส้นทางและชื่อของสคริปต์
4. เซิร์ฟเวอร์วิเคราะห์ว่าเป็นสคริปต์ไม่ใช่เอกสารไฮเปอร์เท็กซ์
5. เซิร์ฟเวอร์ เตรียมสภาพแวดล้อมสำหรับการทำงานและสั่งให้สคริปต์ทำงาน
6. สคริปต์ทำงานอ่านค่าอินพุตและค่าของระบบที่เตรียมไว้จากการทำงานในข้อที่ 5
7. สคริปต์ส่ง MIME¹ ที่เหมาะสมไปทาง STDOUT ก่อนจะส่งเนื้อหาของเอกสารต่อไปให้กับไคลเอ็นท์
8. สคริปต์ส่งส่วนที่เหลือของเนื้อหาทาง STDOUT แล้วหยุดการทำงาน
9. เซิร์ฟเวอร์ สังเกตว่าสคริปต์ทำงานเสร็จสิ้นแล้วจึงทำการยกเลิกการติดต่อ
10. โปรแกรมบราวเซอร์แสดงผลการทำงานของงานที่ได้รับจากสคริปต์

2.2.3 การทำงานร่วมกันระหว่าง CGI และฟอร์ม HTML

การทำงานร่วมกันระหว่าง CGI และ HTML นั้นทำได้โดยการใช้ฟอร์มของ HTML ในการรับอินพุตจากผู้ใช้แล้วส่งไปให้กับ CGI สคริปต์เพื่อทำงานแล้วแสดงผลลัพธ์ในรูปแบบของเอกสาร HTML กลับมา

ภาพประกอบ 2.4 เป็นตัวอย่างฟอร์มเพื่อให้ผู้ใช้ลงทะเบียนสมัครเป็นสมาชิกของระบบ โดยผู้ใช้จะต้องใส่ข้อมูลลงไปในพื้นที่ username, passwd1, passwd2 และ email เมื่อคลิกปุ่ม "Register Me" จะส่งข้อมูลทั้งหมดไปให้ CGI โดยเรียกใช้ CGI สคริปต์ ที่มีชื่อโปรแกรม m_passwd.cgi

การกำหนดชื่อโปรแกรม m_passwd.cgi ซึ่งอยู่ภายใต้ไครคทอรี /cgi-bin/logical-diary ของโฮสต์ pangha.cs.psu.ac.th ซึ่งมี HTTP เซิร์ฟเวอร์ทำงานอยู่ให้เป็นสคริปต์เพื่อที่จะทำงานภายในฟอร์มของเอกสาร HTML มีคำสั่งกำหนดได้ดังนี้

```
<form action="http://pangha.cs.psu.ac.th/cgi-bin/logical-diary/m_passwd.cgi"
method="POST" name="register" onsubmit="return CheckRegister()">
```

¹ มาจากคำว่า Multipurpose Internet Mail Extensions

Register - Netscape

http://pangha.ct.psu.ac.th/~arnudee/WBLD/

Web-Based Logical Diaries

Member Entry

User Registration for Web-Based Logical Diaries

Register Now!

Username: (20 chars, no space) ← username

Password: ← passwd1

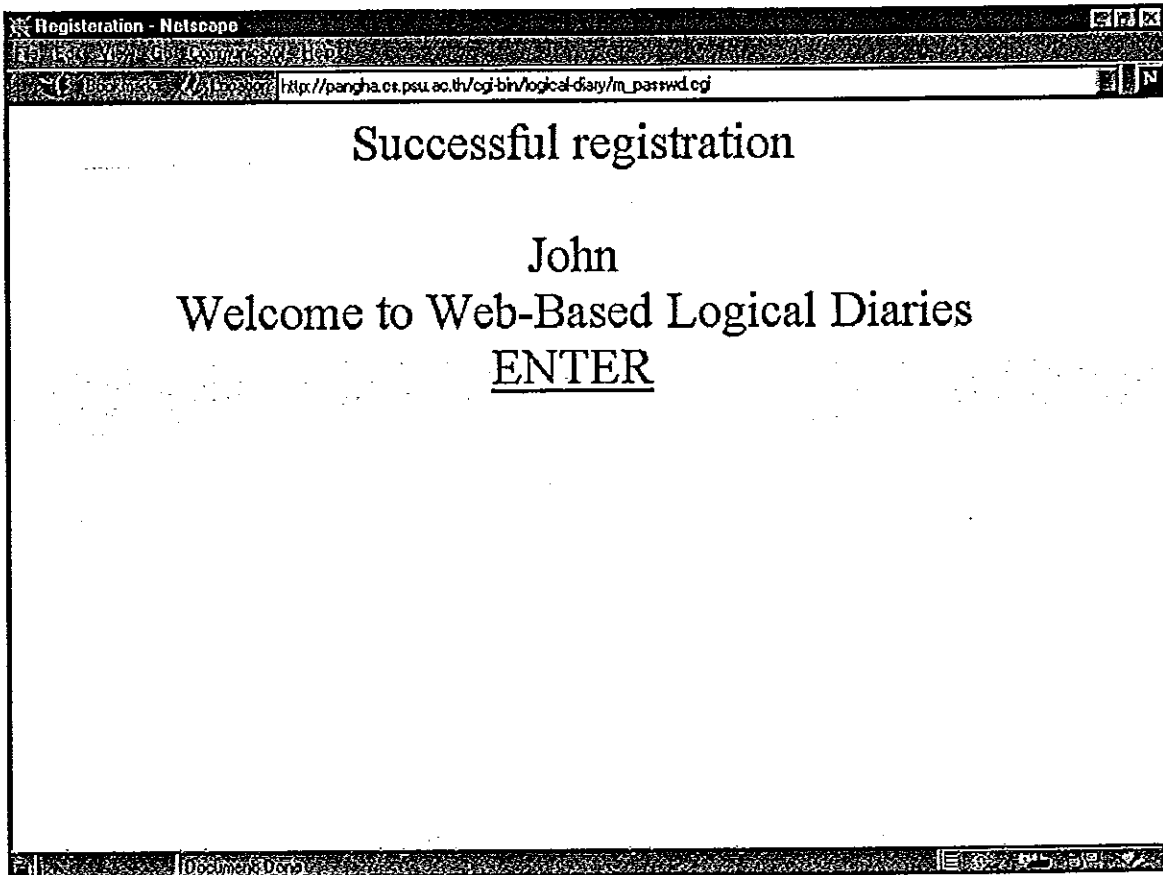
Repassword: ← passwd2

E-mail address: ← email

ภาพประกอบ 2.4 แสดงหน้าจอแรกของระบบลอจิกไดอารี่บนเว็บ
เพื่อเข้าสู่ระบบและการลงทะเบียนเป็นสมาชิก

จากภาพประกอบ 2.4 ลูกศรและคำอธิบายชี้บอกชื่อฟิลด์ของแต่ละข้อมูล ข้อมูลอินพุตทั้งหมดนั้นจะถูกส่งไปใน CGI เป็นคู่ลำดับของชื่อฟิลด์และค่าของมันในรูปดังนี้ username="John" passwd1="john123" passwd2="john123" และ email="SJohn@hotmail.com"

สคริปต์ m_passwd.cgi จะดึงค่าของฟิลด์เหล่านั้นจาก STDIN ของ CGI มาใช้งานโดยใช้ฟังก์ชันของ Pillow Library ช่วยเนื่องจาก Pillow Library เป็นไลบรารีของภาษาเชิงตรรกทำให้สะดวกและง่ายต่อการเขียนโปรแกรมเชิงตรรกยิ่งขึ้น ภายหลังจากการทำงาน m_passwd.cgi จะแสดงผลการทำงานของการทำงานให้ผู้ใช้งานในรูปแบบของเอกสาร HTML โดยการใช้ Pillow Library ในการเขียนเอกสาร HTML เช่นเดียวกัน



ภาพประกอบ 2.5 แสดงหน้าจอผลลัพธ์การทำงานของสคริปต์ m_passwd.cgi

สคริปต์ m_passwd.cgi ใช้ PillowLibrary¹ ในการสร้างเอกสาร HTML ดังภาพประกอบ 2.5 โค้ดในการสร้างเอกสารดังกล่าว โดย PILLOW Library สามารถเขียนได้ดังนี้

```
output_html([ cgi_reply,
              start, title('Registration'),
              begin(body, []), begin(p, [align="center"]),
              begin(font, [size="10"]),
              'Successful registration',
              end(font), end(p),
              begin(p, [align="center"]),
              begin(font, [size="10"]),
              User, \\, & User เป็นตัวแปรเก็บชื่อของผู้ใช้ขณะนั้น
              'Welcome to Web-Based Logical Diaries', \\,
              ref('http://pangha.cs.psu.ac.th/~amrudee/WBLD
              /html/DiaryFrame.htm', 'ENTER'),
              end(font), end(p), end(body),
              end]).
```

โค้ด Pillow Library ข้างต้นสามารถสร้างโค้ด HTML ได้ดังนี้

```
<html>
<title>Registration</title>
<body>
<p align="center"><font size="10">
Successful registration</font></p>
<p align="center"><font size="10">John
<br>Welcome to Web-Based Logical Diaries<br>
<a href="http://pangha.cs.psu.ac.th/~amrudee/WBLD/html/DiaryFrame.htm"> ENTER</a></font></p>
</body>
</html>
```

2.2.4 ระบบความปลอดภัยของเว็บเซิร์ฟเวอร์

เครือข่ายใยแมงมุมเป็นเครือข่ายที่ให้บริการด้านข้อมูลข่าวสารเป็นจำนวนมาก ข้อมูลบางอย่างเป็นความลับและต้องการความปลอดภัย โดยเฉพาะการอนุญาตให้ถึง CGI สคริปต์ทำงาน เพราะต้องใช้ทรัพยากรของเซิร์ฟเวอร์ ดังนั้นควรจะเป็นผู้ใช้ที่เกี่ยวข้องและเหมาะสมเท่านั้นที่มีสิทธิ์จะเข้าไปดูข้อมูลและสั่งให้ CGI สคริปต์ทำงานได้ การทำงานของระบบความปลอดภัยของเว็บเซิร์ฟเวอร์ สามารถทำได้โดยกำหนดไคลเรคทอรี (pre-directory) หรือไฟล์พิเศษในการอนุญาตให้เฉพาะผู้ใช้ที่มีสิทธิ์เท่านั้น หรือการกำหนดกฎเกณฑ์การทำงานของเว็บเซิร์ฟเวอร์ ให้เป็นไปตามความต้องการขององค์กร

หน้าจอที่แสดงในภาพประกอบ 2.6 นั้นจะเกิดขึ้นเมื่อผู้ใช้งานไปยังแฟ้มข้อมูลในไคลเรคทอรีที่มีการกำหนดให้ผู้ใช้ที่มีสิทธิ์เท่านั้นในการร้องขอข้อมูล ไม่ใช่ผู้ใช้ทุกคนสามารถจะเข้าไปได้ ในที่นี้คือการคลิกที่ Member Entry โปรแกรมบราวเซอร์จะแสดงโค๊ดบล็อกบล็อกเพื่อสอบถามชื่อและรหัส

การควบคุมการเข้าถึงข้อมูล (Access Control) หมายถึง การกำหนดกฎการเข้าถึงไฟล์และไคลเรคทอรีที่อยู่ภายในไคลเรคทอรีของเว็บเซิร์ฟเวอร์ ที่ถูกจำกัดสิทธิ์การเข้าใช้งานของผู้ใช้ที่มีสิทธิ์ ตัวอย่างเช่นคุณสามารถอ่านไฟล์พวกนี้ได้ถ้าคุณเข้าสู่ระบบจากระบบเครือข่ายภายในเท่านั้น หรือคุณสามารถควบคุมการเข้าถึงไฟล์ของผู้ใช้แต่ละคนโดยการกำหนดไฟล์พิเศษสำหรับเก็บชื่อและรหัสของผู้ใช้

การตรวจสอบผู้ใช้ที่มีสิทธิ์ (Authentication) คือ ขบวนการตรวจสอบว่าผู้ใช้มีสิทธิ์ในการเข้าถึงแฟ้มข้อมูลโดยการระบุชื่อและรหัส เมื่อเว็บเซิร์ฟเวอร์มีการตรวจสอบว่าเป็นผู้ใช้ที่ถูกต้องก็อนุญาตให้เข้าถึงแฟ้มข้อมูลได้

Register - Netscape

http://panpha.cs.psu.ac.th/~amrudee/WBLD/

Web-Based Logical Diaries

Member Entry

User Registration for Web-Based Lo

Register Now!

Username: (20 chars, no space)

John

Password:

Repassword:

E-mail address:

SJohn@hotmail.com

Register Me Clear

Username and Password Required

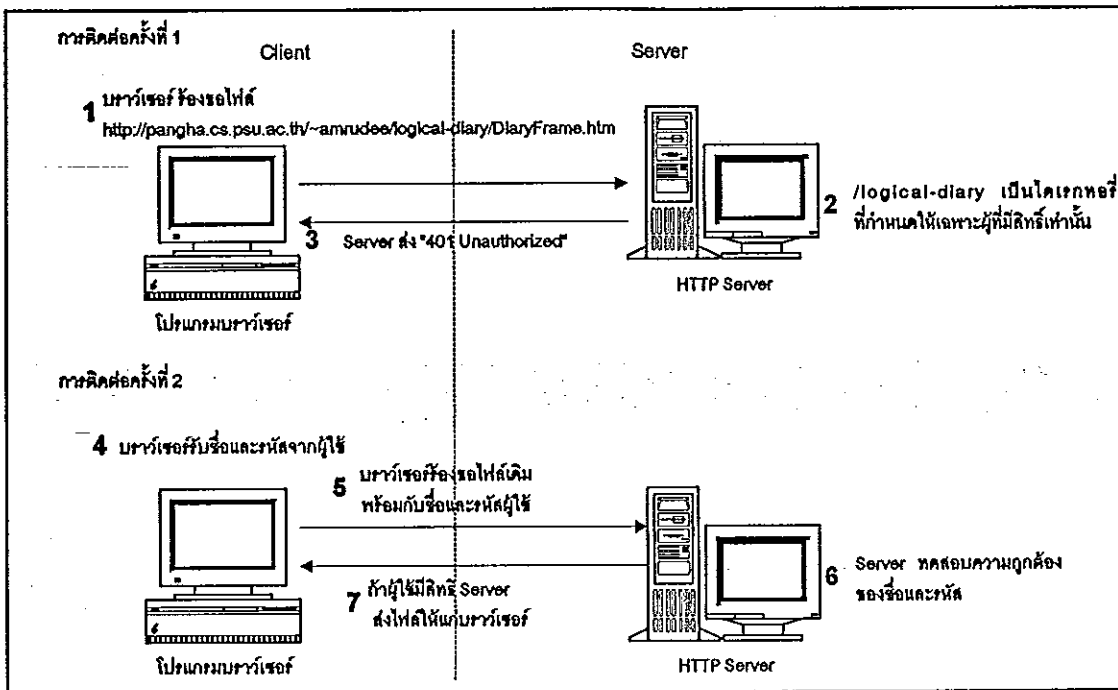
Enter Username and Password (Default Password: admin)

Username: John

Password: *****

OK Cancel

ภาพประกอบ 2.6 แสดงไดอะล็อกบ็อกเพื่อสอบถามชื่อและรหัสเมื่อคลิกที่ Member Entry



ภาพประกอบ 2.7 แสดงการทำงานระหว่างบราวเซอร์กับเว็บเซิร์ฟเวอร์ ในการทำ Authentication

ขั้นตอนการทำงานการร้องขอไฟล์ที่อยู่ภายใต้ไดเรกทอรีที่กำหนดผู้มีสิทธิ์เข้าใช้

1. ผู้ใช้โปรแกรมบราวเซอร์ร้องขอแเพิ่มข้อมูลที่อยู่ภายในไดเรกทอรีที่ถูกป้องกันไว้จากภาพประกอบ 2.7 คือ <http://pangha.cs.psu.ac.th/~amrudee/logical-diary/DiaryFrame.htm>
2. เซิร์ฟเวอร์พบว่าเป็นการเพิ่มข้อมูลที่อยู่ในไดเรกทอรีพิเศษที่ถูกป้องกันสำหรับผู้มีสิทธิ์เท่านั้น
3. เซิร์ฟเวอร์ส่งข้อความกลับไปว่าต้องการ Authentication Required
4. โปรแกรมบราวเซอร์แสดงไดอะล็อกให้ผู้ระบุชื่อและรหัส (ภาพประกอบ 2.6)
5. บราวเซอร์ทำการร้องขอเพิ่มข้อมูลเดิมอีกครั้ง พร้อมส่งชื่อและรหัสส่งไปในคำสั่งข้อความที่ร้องขอ
6. เซิร์ฟเวอร์ตรวจสอบชื่อและรหัสว่าเป็นผู้มีสิทธิ์จริงหรือไม่จากเพิ่มข้อมูลที่เก็บชื่อและรหัสของผู้มีสิทธิ์ (เพิ่ม .htpasswd)
7. ถ้าตรวจสอบแล้วเป็นผู้ที่มีสิทธิ์จริงเซิร์ฟเวอร์จะส่งเพิ่มข้อมูลที่ร้องขอลงกลับไปให้

การควบคุมการเข้าถึงข้อมูลแบ่งเป็น 2 วิธีคือ Global และ Pre-directory วิธี Global เป็นการควบคุมทั้งระบบการทำงานเมื่อเซิร์ฟเวอร์เริ่มทำงาน มีเพิ่มที่สำคัญในการทำงานคือ

access.conf httpd.conf และ srm.conf ซึ่งเจ้าหน้าที่ระบบจะเป็นผู้ดูแล และกำหนดกฎเกณฑ์เงื่อนไขในการเข้าใช้งานของผู้ใช้เว็บเซิร์ฟเวอร์ทั้งระบบ วิธี Pre-directory ผู้ใช้ที่มีสิทธิ์สร้างเอกสารภายในเซิร์ฟเวอร์ให้เป็นไปตามที่ผู้ดูแลระบบเป็นผู้กำหนดขึ้น โดยการสร้างไฟล์ .htpasswd และ .htaccess ผู้ใช้ในระบบทุกคนมีสิทธิ์ในการสร้างไฟล์นี้ไว้ในไดเรกทอรีที่ตนมีสิทธิ์ทำงานเพื่อกำหนดเฉพาะผู้มีสิทธิ์เข้าใช้งานเพิ่มข้อมูลในไดเรกทอรีนั้น

```
AuthUserFile ../.htpasswd
AuthName Web-Based Logical Diaries
AuthType Basic

<Limit GET POST>
require valid-users
</Limit>
```

ภาพประกอบ 2.8 แสดงแฟ้ม .htaccess

โดยที่

AuthUserFile : บอกตำแหน่งและชื่อไฟล์ที่เก็บชื่อและรหัสของผู้ใช้ที่มีสิทธิ์ (.htpasswd)

AuthName : เก็บชื่อหรือข้อความที่ต้องการให้แสดงบอกผู้ใช้ว่าเป็นกลุ่มผู้ใช้งานอะไร

AuthType : บอกชนิดของการทำ Authentication ในที่นี้เป็นชนิด Basic

<Limit GET POST> : อนุญาตให้เฉพาะผู้ใช้ที่มีสิทธิ์เข้าใช้ทำโอเปอเรชัน GET และ POST

require valid-users : ผู้ที่มีสิทธิ์คือผู้ใช้ที่ถูกต้องเท่านั้นจึงเข้าถึงข้อมูลได้

ในที่นี้สมมติให้แฟ้ม .htaccess นี้อยู่ในไดเรกทอรี /opt/http/cgi-bin/logical-diary/script

และภายในไดเรกทอรี script เก็บโปรแกรม CGI Script สำหรับการทำงานของลอจิกไดอารีบนเว็บทั้งหมด ดังนั้นผู้ใช้ที่มีสิทธิ์จะใช้งาน CGI Script ของ Web-Based Logical Diary จะต้องเป็นผู้ใช้ที่มีสิทธิ์ซึ่งหมายถึงได้ทำการสมัครเป็นสมาชิกของระบบแล้ว

```
Jack:Gj6I/dj8Xq2SI
Jenny:opUJt77OG/OVc
John:I/AhLrpQ3YJU
```

ภาพประกอบ 2.9 แสดงแฟ้ม .htpasswd

ฝ่ายหอสมุด
ศูนย์นิเทศฯ อรรถกวีสุนทร

ไฟล์ .htpasswd เก็บชื่อของผู้ใช้ที่มีสิทธิ์และ password ที่ได้มีการเข้ารหัสไว้เพื่อไม่ให้
อ่านออกได้โดยใช้ฟังก์ชัน crypt() ของภาษา C ในส่วนนี้ต้องใช้ภาษาซีในการทำงานไม่สามารถ
จะใช้ BinProlog ในการเขียนได้

จากภาพประกอบ 2.9 ในที่นี้มีผู้ที่สามารถเข้าใช้งานใน Web-Based Logical Diaries ได้
3 คนคือ Jack Jenny และ John เท่านั้น ผู้ใช้คนอื่นๆ ไม่มีสิทธิ์

2.2.5 จาวาสคริปต์ (Java Script)

จาวาสคริปต์ คือ ภาษาสคริปต์เชิงวัตถุ (object-based Scripting language) พัฒนาขึ้น
โดยบริษัท Netscape เพื่อใช้ร่วมกับ Netscape Navigator ซึ่งมีการทำงานแบบไคล์เอ็นท์/
เซิร์ฟเวอร์ โดย Netscape Navigator จะทำการแปลงประโยคคำสั่งของจาวาสคริปต์ที่ฝังรวมมา
กับเอกสารไฮเปอร์เท็กซ์ที่เซิร์ฟเวอร์ส่งมาให้เพื่อทำปฏิบัติการทางด้านไคล์เอ็นท์ แต่ในขณะนี้
โปรแกรมบราวเซอร์ตัวอื่นๆ เช่น Internet Explorer ก็สามารถจะทำงานร่วมกับ จาวาสคริปต์ ได้
เช่นกัน จาวาสคริปต์ มี 2 ชนิดคือ Navigator จาวาสคริปต์ หรือเรียกว่า จาวาสคริปต์ทำงานที่
ไคล์เอ็นท์ (Client-Side Java Script) และ Live Wire จาวาสคริปต์ หรือเรียกว่า จาวาสคริปต์ทำงาน
ที่เซิร์ฟเวอร์ (Server Side Java Script)

การใช้จาวาสคริปต์ทำให้เอกสารไฮเปอร์เท็กซ์ที่สร้างขึ้นมีประสิทธิภาพในการทำงาน
เพิ่มขึ้น มีสีสัน ไม่เป็นเพียงแต่การแสดงผลเอกสารธรรมดาเท่านั้น สามารถทำงานร่วมกับฟอร์มของ
HTML เพื่อตรวจสอบข้อผิดพลาดของข้อมูลก่อนจะทำการส่งไปให้ เซิร์ฟเวอร์ ประมวลผล

การฝังจาวาสคริปต์ร่วมกับเอกสาร HTML มี 2 วิธี วิธีที่ 1 คือการเขียนคำสั่งและ
ฟังก์ชันต่างๆ ภายใต้อคริปต์มีคำสั่งการเขียนดังนี้

```
<SCRIPT LANGUAGE="JavaScript">
    <!-- Java Script statement ...
    -->
</SCRIPT>
```

วิธีที่ 2 คือการตรวจสอบเหตุการณ์ที่เกิดขึ้นกับออปเจกต์ต่างๆ ในเอกสารโดยการฝัง
ร่วมกับ HTML tag อื่นๆ ซึ่งส่วนใหญ่จะเป็นพวกออปเจกต์ของฟอร์ม ซึ่งเหตุการณ์การทำงาน
จะเป็นเหตุการณ์ที่เกิดขึ้นกับออปเจกต์ต่างๆ ในฟอร์มเช่นการคลิกปุ่ม การพิมพ์ข้อความ

```
<form action="http://pangha.cs.psu.ac.th/cgi-bin/logical-diary/m_passwd.cgi"
method="POST" name="register" onsubmit="return CheckRegister()">
```

เอกสาร HTML ในภาพประกอบ 2.4 มีการฝังประโยคคำสั่ง JavaScript ไว้ในโค้ดของเอกสาร เมื่อผู้ใช้คลิกปุ่ม “Register Me” จะเรียกใช้ไพรซีเคอร์ CheckRegister เพราะได้ใส่คำสั่ง onsubmit = “return CheckRegister()” ตามโค้ดที่แสดงข้างบน เพื่อตรวจสอบอินพุตก่อนที่จะส่งคำสั่งเพื่อรัน CGI Script m_passwd.cgi หน้าที่ของไพรซีเคอร์ CheckRegister คือตรวจสอบอินพุตว่าผู้ใช้ใส่ข้อมูลครบถ้วนแล้วหรือยัง และใส่ชื่อ และ รหัส ถูกต้องหรือไม่ ถ้าถูกต้องหมดจะส่งค่ากลับเป็นจริงเพื่อทำงานต่อไป ถ้าไม่ก็จะส่งค่ากลับเป็นเท็จเพื่อหยุดการทำงานและแสดงข้อผิดพลาดให้ผู้ใช้ทราบ

โค้ดแสดง จาวาสคริปต์ ที่ฝังอยู่ในฟอร์มการลงทะเบียนของภาพประกอบ 2.4

```
<Script language="JavaScript"><!--
var ValidData=true //ประกาศตัวแปรโกลบ
function CheckRegister()
(
  usernamecheck() //ตรวจสอบฟิลด์ username ว่ามีช่องว่างหรือไม่
  passwordcheck() //ตรวจสอบฟิลด์ passwd1 มีค่าเท่ากับ passwd2 หรือไม่
  emailcheck() // ตรวจสอบฟิลด์ email ว่ามีสัญลักษณ์ @ หรือไม่
  TValidData=ValidData // ให้ TValidData มีค่าเหมือนค่าของ ValidData
  ValidData=true // ให้ ValidData มีค่าเป็นจริงเพื่อการตรวจสอบครั้งใหม่
  if (TValidData) return true// ถ้าไม่มีข้อผิดพลาดของอินพุตจะส่งค่ากลับ
  else return false // เป็นจริง ไม่เช่นนั้นจะส่งค่ากลับเป็นเท็จ
)
// --><Script>
```


โค้ดฟังก์ชัน usernamecheck()

```
function usernamecheck()
{
var string1=document.register.username.value
    if (string1=="") // ถ้าค่าของ username เป็นช่องว่างแสดงข้อความผิดพลาด
        { alert("Please enters username field!")
          ValidData=false
        }
    if (string1.indexOf(" ")!=-1)// ถ้ามีช่องว่างใน username แสดงข้อความผิดพลาด
        { alert("Don't have space in the username!")
          document.register.username.focus()
          ValidData=false
        }
    if (string1.length > 20)// ถ้า username มีความยาวมากกว่า 10 อักขร
        { alert("Username has length more than 10!")
          document.register.username.focus()
          ValidData=false
        }
}
```

บทที่ 3

โพรลอค

โพรลอค (Programming in Logic = Prolog) คือ ภาษาสั่งงานคอมพิวเตอร์เชิงตรรกะที่ทำงานกับสัญลักษณ์เชิงตรรกะ เหมาะสำหรับการคำนวณตัวเลขทางคณิตศาสตร์ที่ไม่ซับซ้อนและการแก้ปัญหาซึ่งรวมไปถึงการกำหนดข้อมูลที่เป็นข้อเท็จจริง และความสัมพันธ์ต่างๆ โดยใช้หลักเกณฑ์โดยทั่วไปทางตรรกวิทยา

¹ภาษาโพรลอคเป็นการผสมผสานความคิดต่างๆ เข้าด้วยกันจากการทำงานที่ใช้

- (1) ประโยคฮอร์น (Horn clause) เพื่อแสดงความรู้ที่มีความสัมพันธ์เชิงตรรกะ
- (2) สไตล์การเขียนโปรแกรมแบบ ปรายละเอียดของเนื้อหา (descriptive style)
- (3) อรรถาธิบาย (declarative) และ ประมวลผลความหมาย (procedural semantics)
- (4) การประมวลผลลิสต์ (List processing)
- (5) ความสามารถในการเชื่อมโยงคำสั่งระดับ meta ให้เข้ากับระดับออปเจกต์

```
parent(pam,bob).      % Pam is parent of Bob
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).

female(pam).         % Pam is female
female(liz).
female(pat).
female(ann).
male(tom).           % Tom is male
male(bob).
male(jim).

grandparent(X,Z):-  % X is grandparent of Z if
    parent(X,Y),   % X is parent of Y and
    parent(Y,Z).   % Y is a parent of Z
predecessor(X,Z):- % Rule pr1: X is a predecessor of Z
    parent(X,Z).
predecessor(X,Z):- % Rule pr2: X is a predecessor of Z
    parent(X,Y),
    predecessor(Y,Z).
```

ภาพประกอบ 3.1 แสดงโค้ดโปรแกรมครอบครัวที่เขียนโดยภาษาโพรลอค

¹การเขียนโปรแกรมเชิงตรรกะ ผ.ศ.อิสระ ศรีสินทร หน้า 1-4

3.1 ประโยคและลักษณะทั่วไปของการเขียนภาษาโปรแกรม

โปรแกรมภาษาโปรแกรมประกอบด้วยประโยค (clause) โดยแต่ละประโยคต้องจบลงด้วยจุดประโยคโปรแกรมแบ่งเป็น 3 ชนิดคือ ข้อเท็จจริง (facts) กฎ (rules) และ คำถาม (goal)

ข้อเท็จจริง คือประกาศสิ่งที่เป็นจริงเสมอ ความจริงที่ไม่มีเงื่อนไข
ตัวอย่าง ข้อเท็จจริง

```
parent(pam,bob).
```

เป็นข้อเท็จจริงแสดงความสัมพันธ์ของครอบครัวว่า pam เป็นผู้ปกครองของ bob โดย parent คือชื่อของความสัมพันธ์ pam และ bob เป็นอาร์กิวเมนต์ของความสัมพันธ์ เป็นข้อมูลชนิดอะตอม ข้อมูลชนิดอะตอมคือหน่วยข้อมูลที่เล็กที่สุดขึ้นต้นด้วยอักษรภาษาอังกฤษตัวเล็ก หรืออักษรภาษาอังกฤษตัวใหญ่ที่อยู่ภายใต้เครื่องหมายฟันทอง ('...') เช่น 'Pam'

การเขียนคอมเมนต์อธิบายโปรแกรมเขียนหลังเครื่องหมาย "%" สำหรับคอมเมนต์ 1 บรรทัด และ เขียนภายใต้เครื่องหมาย /* คอมเมนต์ */ สำหรับหลายบรรทัด เรียกประโยคในตัวอย่างว่าเพรตดิเคต parent/2(+,+) มีความหมายว่าประกอบด้วย 2 อาร์กิวเมนต์เครื่องหมายบวก แสดงว่าเป็นอินพุต ถ้าเครื่องหมายลบแสดงว่าเป็นเอาต์พุตของเพรตดิเคต

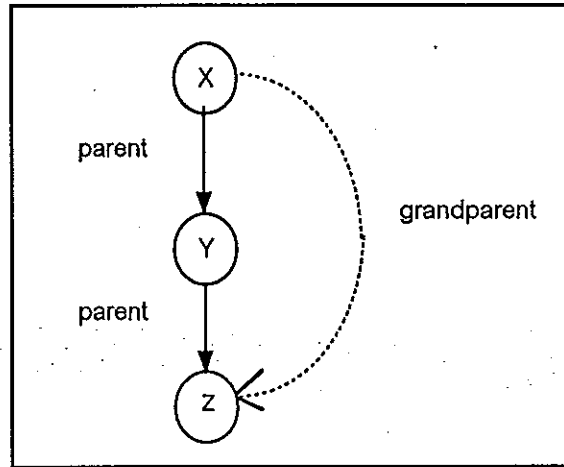
กฎ คือประกาศสิ่งที่เป็นจริงภายใต้เงื่อนไขที่กำหนดให้

ตัวอย่าง กฎ

```
grandparent(X,Z) :- parent(X,Y), parent(Y,Z). % X is grandparent of Z if
                                     % X is parent of Y and Y is parent of Z
```

กฎกำหนดว่า X เป็นปู่ของ Y ก็ต่อเมื่อ X เป็นผู้ปกครองของ Y และ Y เป็นผู้ปกครองของ Z โดย X และ Z เป็นตัวแปรในโปรแกรม ตัวแปรในโปรแกรมขึ้นต้นด้วยอักษรภาษาอังกฤษตัวพิมพ์ใหญ่เสมอเครื่องหมาย ";" (comma) หมายถึงเงื่อนไขทั้งสองต้องเป็นจริงจึงจะทำให้กฎข้อนี้เป็นจริง

กฎประกอบด้วย 2 ส่วน คือ head หรือ ข้อสรุปของกฎอยู่ทางซ้ายมือ (conclusion part) และ body หรือ เงื่อนไข (condition part) อยู่ทางขวามือ



ภาพประกอบ 3.2 แสดงความสัมพันธ์ระหว่างเพรคติกัด grandparent/2 และ parent/2

คำถาม หรือ เป้าหมาย คือการสอบถามข้อเท็จจริงในโปรแกรมโพรลอคสามารถตอบ “yes” หรือ “no” ไปจนถึงค่าของตัวแปรข้อมูลที่สอดคล้องกับคำถามที่ค้างขึ้นและมีความซับซ้อนได้ง่าย

เมื่อทำการโหลดโปรแกรมลงสู่ระบบโพรลอคจะปรากฏพอร์มของระบบ (โพรลอคส่วนใหญ่ใช้สัญลักษณ์ “?-” เป็นพอร์มของระบบ) เพื่อให้ผู้ใช้สอบถามในที่นี้สมมติว่าโหลดตัวอย่างโปรแกรมครอบครัวจากภาพประกอบ 3.1 ลงสู่หน่วยความจำเรียบร้อยแล้ว

ตัวอย่าง คำถาม

?-parent(bob,pat).	% คำถาม bob เป็นผู้ปกครองของ pat ใช่ไหม
yes	% คำตอบ ใช่ เพราะในโปรแกรมมีข้อเท็จจริงนี้อยู่

คำถาม parent(bob,pat) จะตรงกับข้อเท็จจริงในโปรแกรม คือ parent(bob,pat) เรียกการทำงานอย่างนี้ว่า pattern matching เทอม 2 เทอมจะ match กันได้เมื่อทั้ง 2 เทอมเหมือนกันทุกประการ หรือ ตัวแปรในทั้งสองเทอมสามารถถูกแทนค่า โดยค่าของเทอมอีกเทอมหนึ่งได้ภายหลังจากการแทนค่าทั้ง 2 เทอมจะเหมือนกันทุกประการ

ตัวอย่าง คำถาม

?-parent(liz,pat).	% คำถาม liz เป็นผู้ปกครองของ pat ใช่ไหม
no	% คำตอบ ไม่ใช่ เพราะในโปรแกรมไม่มีข้อเท็จจริงนี้อยู่

นอกจากการตอบคำถาม “yes” หรือ “no” โพรลอคสามารถถามคำถามที่ให้คำตอบเป็น
อย่างอื่นด้วย

ตัวอย่าง คำถาม

?-parent(X,liz).	% คำถาม ใครเป็นผู้ปกครองของ liz
X=tom	% เมื่อโพรลอคทำ pattern matching โดยแทนค่า X ด้วย tom
	% ทำให้ประโยคคำถามนั้นมีค่าเป็นจริง
?-parent(bob,X).	% คำถาม bob เป็นผู้ปกครองของใคร
X=ann;	% เมื่อโพรลอคทำ pattern matching จะได้ X=ann
	% แต่คำถามนี้มี 2 คำตอบ
	% ดังนั้นเมื่อผู้ใช้ต้องการทราบคำตอบต่อไปก็พิมพ์เครื่องหมาย “;”
	% โพรลอคจะทำการย้อนรอยกลับไปค้นหาข้อเท็จจริงต่อไป ที่ทำให้คำถาม
	% เป็นจริง เรียกการทำงานนี้ว่า backtracking
X=pat;	% ถ้าไม่สามารถจะหาคำตอบได้อีกโพรลอคจะตอบ “no” เพราะการ
no	% backtracking กลับไปอีกครั้งนั้นไม่สามารถทำให้เป้าหมายเป็นจริงได้อีก

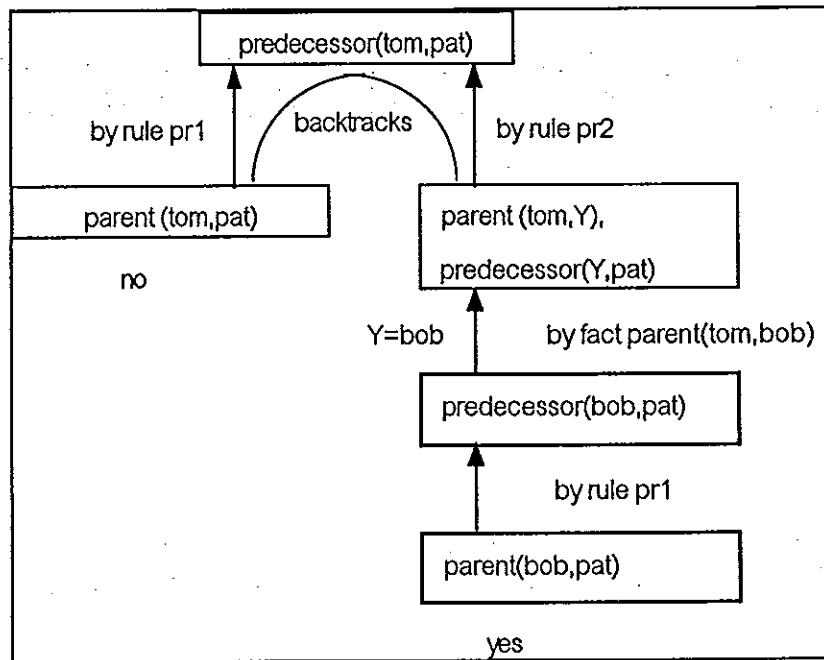
ตัวอย่าง คำถาม predecessor ด้านล่างต่อไปนี้เป็นตัวอย่างที่แสดงให้เห็นการทำงานของ
ของการย้อนรอยที่ซับซ้อนขึ้นของโพรลอค

โพรลอคสามารถสร้างข้อเท็จจริงใหม่ได้ โดยการใช้ข้อเท็จจริงและกฎที่มีเขียนไว้ใน
โปรแกรมก่อนเพื่อเกิดเป็นข้อเท็จจริงใหม่

?-predecessor(tom,pat).
yes

เริ่มต้นการทำงานเป้าหมาย predecessor(bob,pat) จะถูกแทนด้วยเป้าหมายใหม่ คือ
parent(bob,pat) โดยกฎที่ 1 คือ predecessor(X,Y) :- parent(X,Y) เป้าหมาย คือ
predecessor(bob,pat) ตรงกับกฎ predecessor(X,Y) เพราะส่วนหัวของกฎเหมือนกับเป้าหมาย
(match) ดังนั้น X จะมีค่าเท่ากับ bob และ Y จะมีค่าเท่ากับ pat แต่ parent(bob,pat) นั้นเป็นเท็จ ดังนั้นโพรลอคจึงย้อนรอย (backtracks) ไปยังเป้าหมายเริ่มต้นเพื่อใช้กฎอื่นต่อไป ดังนั้นกฎ pr2 ก็ถูก
ทำต่อไป ดังนั้นเป้าหมาย predecessor(tom,pat) จะถูกแทนที่ด้วย 2 เป้าหมายคือ

$\text{parent}(\text{tom}, Y), \text{predecessor}(Y, \text{pat})$ โพรลอคจะทำงานตามลำดับเป้าหมายที่เขียนไว้เมื่อเริ่มทำ $Y=\text{bob}$ และ $\text{predecessor}(\text{bob}, \text{pat})$ ดังนั้น กฎ pr1 จะถูกนำมาใช้งานอีก โดยโพรลอคใช้เช็คของ ตัวแปรตัวใหม่สำหรับกฎที่ถูกเรียกใช้งานแต่ละครั้ง ดังนั้นเป้าหมายนี้ถูกแทนด้วย $\text{parent}(\text{bob}, \text{pat})$ และผลปรากฏว่ามีข้อเท็จจริงนั้น ดังนั้นโพรลอคจึงตอบคำถามเป็น “yes” และหยุดการทำงาน



ภาพประกอบ 3.3 แสดงขั้นตอนการ backtrack เพื่อค้นหาคำตอบ predecessor(tom, pat).

ในกรณีที่ไม่ต้องการให้โพรลอคทำการย้อนรอยกลับไปทดสอบกฎถัดไปสามารถใส่คำสั่ง **ตัด (cut)** ลงในกฎได้ สัญลักษณ์ของคำสั่งคือ “!” การใช้คำสั่งทำให้ประหยัดเวลาในการประมวลผล หรือย้อนรอยเพื่อทำงานต่อไปของระบบ

ตัวอย่าง โปรแกรมการหาค่าสูงสุด $\text{max}/3(X+, Y+, \text{Max}-)$ มีโปรแกรมดังนี้
 โค้ดของเพรดิเคต $\text{max}/3(+, +, -)$ กรณีไม่ใช้ตัด

$\text{max}(X, Y, X) :- X \geq Y.$	% ไม่ใช้ตัด ถ้า X มากกว่าหรือเท่ากับ Y ให้ส่งค่า X กลับไป
$\text{max}(X, Y, Y) :- X < Y.$	% ถ้า X น้อยกว่า Y ให้ส่ง Y กลับไป

โค้ดของเพรดิเคต $\text{max}/3(+, +, -)$ กรณีใช้ตัด

```

max(X,Y,X) :- X >= Y,!.    % ถ้า X มากกว่าหรือเท่ากับ Y ให้ส่งค่า X กลับไป
max(X,Y,Y).              % ถ้ากฎด้านบนไม่สำเร็จ ส่งค่า Y กลับไปไม่ต้องมี
                          % การคำนวณอีกครั้ง

```

ภาษาโปรลอคมีการทำงานกับโครงสร้างข้อมูลแบบลิสต์ ประโยคข้อเท็จจริงของโพรลอคคือไปนี้ animal/1 แสดงรายชื่อสัตว์ซึ่งใช้โครงสร้างข้อมูลแบบลิสต์

```

animal([ant,bird,fish,cat,rat,tiger]).

```

[ant,bird,fish,cat,rat,tiger] เป็นการเขียนข้อมูลแบบลิสต์ ลิสต์ เป็นโครงสร้างข้อมูลที่ง่ายและใช้กันมากในการเขียนโปรแกรมที่ไม่ใช่การคำนวณทางตัวเลข (non-numeric programming) ลิสต์ คือ ลำดับของสิ่งของจำนวนใดจำนวนหนึ่ง

โครงสร้างข้อมูลของลิสต์ประกอบด้วย 2 ส่วนคือ

- ข้อมูลตัวแรก เรียกว่า เฮด (head) ของลิสต์
- ข้อมูลส่วนที่เหลือทั้งหมดของลิสต์เรียกว่า เทล (tail)

ลิสต์ว่างแทนด้วยสัญลักษณ์ “[]”

ตัวอย่าง การสอบถามข้อมูลลิสต์

```

?-animal(A).
A = [ant,bird,fish,cat,rat,tiger]

```

ตัวอย่าง การอ้างถึงข้อมูลในลิสต์จะต้องอ้างข้อมูลของส่วนที่เป็นหัวของลิสต์เสมอ

```

?-animal([H|T]).
H = ant
T = [bird,fish,cat,rat,tiger]

```

3.2 BinProlog

BinProlog เป็นคอมไพเลอร์ของภาษาโพรลอกที่พัฒนาขึ้นโดย Tarau 1997¹ สามารถไปดาวน์โหลดมาใช้งานได้ที่ <ftp://clement.info.umoncton.ca> งานวิทยานิพนธ์ฉบับนี้ใช้ BinProlog 5.75 สำหรับเขียน CGI Script บนระบบปฏิบัติการยูนิกซ์ (unix) ซึ่งทำงานบนเครื่อง Sun Solaris การใช้งานส่วนใหญ่จะเหมือนกับโปรแกรมภาษาโพรลอกทั่วไป เหตุผลที่เลือกใช้ BinProlog 5.75 เพราะสามารถทำงานบนระบบปฏิบัติการยูนิกซ์ ทำงานร่วมกับ PiLLow Library ได้ และโปรแกรมที่พัฒนาขึ้นนี้มีการใช้งานร่วมกับโปรแกรมเอเจนท์ไดอารี่บนเว็บของ นางสาวเดือนเพ็ญ กัญจิติธรรม² (Logical Diary Agents on the Web) ซึ่งเขียนโดยใช้ BinProlog เช่นเดียวกัน พร้อมทั้งเอกสารให้ศึกษาการใช้งานพร้อมกับโปรแกรมที่โหลดมาใช้งานโดยไม่ต้องเสียค่าใช้จ่ายใดๆ

ในการทำวิจัยผู้วิจัยพบปัญหาในการคอมไพล์ source ไฟล์ของ BinProlog ไม่เป็นเอ็กซ์เซคิวทีฟไฟล์เพื่อทำงานกับ CGI เพราะเอ็กซ์เซคิวทีฟไฟล์ที่ได้มานั้นไม่สามารถทำงานได้ คำสั่งการสร้างเอ็กซ์เซคิวทีฟไฟล์ของ BinProlog คือเพเรคดิคเตด `make_executable_unix_appl/3(+,+,+)` โดยเทอมที่ 1 คือตำแหน่งและที่อยู่ของคำสั่ง `bp` ซึ่งอาจจะมีการระบุอปชั่นของการคอมไพล์ลงไปด้วยเช่น `-q5` เป็นการระบุว่าเป็นการคอมไพล์การทำงานแบบไม่มีการแสดงข้อความใดๆ ออกมาอยู่ในระดับที่ 5 (มีตั้งแต่ระดับที่ 1 ถึง 5) `-h20000` เป็นการระบุว่าให้จอง hash table สำหรับการทำงานขนาด 20000 byte เทอมที่ 2 คือชื่อและที่อยู่ของ source ไฟล์ เทอมที่ 3 คือชื่อและที่อยู่ของเอ็กซ์เซคิวทีฟไฟล์

ตัวอย่าง การเรียกใช้คำสั่ง `make_executable(+,+,+)` การคอมไพล์ไฟล์ `m_passwd.pl` ไปเป็นเอ็กซ์เซคิวทีฟไฟล์ที่ชื่อ `m_passwd.cgi` ซึ่งอยู่ภายในไดเรกทอรีเดียวกัน และคำสั่ง `bp` อยู่ภายใต้ไดเรกทอรี `/opt/bin` ซึ่ง `bp` เป็นคำสั่งการในการรัน BinProlog

```
?-make_executable_unix_appl('/opt/bin/bp -q5', 'm_passwd.pl', 'm_passwd.cgi').
```

ภายหลังจากการทำงานจะได้เอ็กซ์เซคิวทีฟไฟล์ชื่อ `m_passwd.cgi` แต่ไม่สามารถใช้งานผ่าน CGI ได้ จากการสืบเสาะและวิเคราะห์สาเหตุของปัญหาที่เกิดขึ้นผู้วิจัยพบว่า การตั้งชื่อเอ็กซ์เซคิวทีฟไฟล์ไม่สามารถจะเปลี่ยนชื่อเป็นชื่ออื่นๆ ที่ไม่ใช่ชื่อเดียวกับ source ไฟล์ได้จึงทำให้มีชื่อซ้ำกับ source ไฟล์ถึงแม้ว่าจะมีนามสกุลที่แตกต่างกัน เมื่อโปรแกรมทำงานจึงไปทำการเอ็กซ์เซคิวทีฟตัวเอ็กซ์เซคิวทีฟเองแทนที่จะไปเอ็กซ์เซคิวทีฟ source ไฟล์จึงเกิดปัญหาขึ้นในการทำงาน ผู้วิจัยจึงแก้ปัญหาดังกล่าวโดย

¹ BinProlog Copyright © Paul Tarau 1992-1996 All rights reserved

² ภาวละเวียดเพิ่มเติมในหัวข้อ 4.1.2.2

การระบุชื่อ source ไฟล์หรือมีนามสกุลให้ด้วยในตอนที่ 1 ของเพรคติกต์ ดังนั้นผู้วิจัยได้เปลี่ยนการคอมไพล์เป็นดังนี้และปัญหาดังกล่าวจึงหมดไป

```
?-make_executable_unix_appl('/opt/bin/bp -q5 m_passwd.pl', 'm_passwd.pl', 'm_passwd.cgi').
```

3.3 PiLLoW/CIAO Library สำหรับระบบเครือข่ายหรือระบบเครือข่ายใยแมงมุม

PiLLoW คือ ไลบรารีของภาษาเชิงตรรกที่อำนวยความสะดวกสำหรับการสร้างเอกสาร HTML การสร้างฟอร์ม HTML การจัดการกับฟอร์ม การเข้าและวิเคราะห์เอกสาร HTML และการเข้าไปยังโค้ดที่ระบุในแอดเดรสของ HTML และการทำงานร่วมกับ CGI ผู้พัฒนาคือ Daniel Cabeza, Manuel Hermenegildo และ Sacha Varma เพื่อใช้งานกับระบบ &-Prolog และระบบ CIAO แต่สามารถนำไปปรับปรุงใช้ร่วมกับระบบโปรแกรมเชิงตรรกอื่นๆ ได้ ในที่นี้ผู้ทำวิทยานิพนธ์ได้นำมาใช้งานร่วมกับ BinProlog

3.3.1 การจัดการฟอร์มด้วย PiLLoW

จากภาพประกอบ 2.4 โปรแกรมใช้ PiLLoW ไลบรารีในการดึงค่าอินพุตค่าของ username, passwd1, passwd2 และ email ด้านล่างคือโค้ดโปรแกรมที่ใช้ PiLLoW ไลบรารีในการดึงค่าอินพุตจาก CGI

```
:-['./pillow.pl'].           %โหลด pillow.pl ไลบรารีมาใช้งานในระบบ
:-['./p_passwd.pl'].        %โหลดโปรแกรม p_passwd.pl มาใช้งานในระบบ

main:-
    get_form_input(Input), %ดึงค่าอินพุตทั้งหมดจาก CGI มาอยู่ในลิสต์ของตัวแปร Input
                           %ดึงค่าของ Username มาใส่ในตัวแปร UserId
    get_form_value(Input, username, UserId),
                           %ดึงค่าของ password1 มาใส่ในตัวแปร Pwd1
    get_form_value(Input, password1, Pwd1),
                           %ดึงค่าของ password2 มาใส่ในตัวแปร Pwd2
    get_form_value(Input, password2, Pwd2),
                           %ดึงค่าของ email มาใส่ในตัวแปร Email
    get_form_value(Input, email, Email),
                           %เรียกใช้เพรคติกต์ add_user/4(+,+,+,+) เพื่อเพิ่มชื่อ user ใหม่ลงสู่ระบบ
    add_user(UserId,Pwd1,Pwd2,Email).
```

หน้าที่ของเหตุการณ์ต่างๆที่ใช้งาน

`get_form_input(Input)` ทำหน้าที่แปลงอินพุตจากฟอร์ม (ไม่ว่าจะเป็น Post หรือ Get method หรือ ENCTYPE multipart/form-data) มาอยู่ในลักษณะของคู่ลำดับ แอตทริบิวต์ = ค่าของ แอตทริบิวต์ (`attribute=value`) แปลงแอตทริบิวต์ที่ไม่มีค่าเป็น 'Empty' จากตัวอย่างจะได้ `Input=[username='John', password1='john123', password='john123', email='SJohn@hotmail.com']` ส่วนค่าข้อมูลอื่นๆ อาจจะเป็นอะตอมหรือเลขจำนวน ข้อมูลที่มีค่ามากกว่า 1 บรรทัด (เช่น จากออปเจกต์ text areas หรือจากไฟล์) แปลงให้เป็นลักษณะของบรรทัด

`get_form_value(Input,Var,Val)` ทำหน้าที่ดึงเอาค่าของ Var ที่เป็นของแอตทริบิวต์ใน Input ค่าที่ได้จะอยู่ใน Val กรณีที่ไม่มีแอตทริบิวต์นั้นใน Input ค่า Val จะมีค่าเป็น 'Empty'

ตัวอย่าง

```
get_form_value(Input,username,UserId).
UserId= 'John'
```

นอกจากนั้นยังมีเหตุการณ์ของ PiLLow Library ในการจัดการฟอร์มที่ใช้ในสคริปต์โปรแกรมอื่นๆดังนี้

`form_empty_value(V)` ทำหน้าที่ตรวจสอบว่าค่า V มีค่าเท่ากับ "Empty" หรือไม่ ถ้าใช่จะตอบเป็น yes ถ้าไม่ใช่จะตอบเป็น "no"

`getenvstr('REMOTE_USER',RUser)` ทำหน้าที่ดึงค่าชื่อของผู้ใช้ทางไกลที่เข้ามาใช้ระบบในกรณีที่เข้าใช้เอกสารกำหนดผู้มีสิทธิ์เข้าใช้ จากภาพประกอบ 2.5 ภายหลังจากเข้าสู่ระบบเมื่อเรียกใช้ predicate นี้ในการทำงานจะได้ค่าของ RUser คือ 'John'

3.3.2 การเขียนเอกสาร HTML ด้วย PiLLow Library

จากภาพประกอบ 2.5 แสดงให้เห็นถึงเอกสารที่สร้างโดยใช้ PiLLow Library แสดงได้การทำงานได้ดังนี้

```

output_html ([
    cgi_reply,
    start,
    title('Registration'),
    begin(body, {}),
    begin(p, [align="center"]),
    begin(font, [size="10"]),
    'Successful registration',
    end(font), end(p),
    begin(p, [align="center"]),
    begin(font, [size="10"]),
    User, \\, % User เป็นตัวแปรเก็บชื่อของผู้ใช้งานนั้น
    'Welcome to Web-Based Logical Diaries', \\,
    ref('http://pangha.cs.psu.ac.th/~amrudee/WBLD/
diary-html/DiaryFrame.htm', 'ENTER'),
    end(font), end(p), end(body),
    end]).

```

เหตุการณ์ที่สำคัญในการทำงานคือ output_html(F) ทำหน้าที่ในการส่งผลลัพธ์ซึ่งจะอยู่ในรูป ฟอรัม HTML กลับไปทาง standard output ในที่นี้คือทาง CGI เทอมของ output_html (F) จะเป็นลิทซ์ของเหตุการณ์อื่นๆ ในการสร้าง HTML Tag

ตาราง 3.1 แสดงคำสั่งโปรลอคของ Pillow ไลบรารีในการสร้างเอกสาร HTML

คำสั่งของ Pillow ไลบรารี	HTML ที่ได้	หมายเหตุ
cgi_reply	Content-type:text/html	
start	<html>	
end	</html>	
begin(name,Atts)	< name Atts>	โดยที่ Atts เป็นลิทซ์ของแอตทริบิวต์ซึ่งอยู่ในรูปคู่ลำดับของชื่อแอตทริบิวต์=ค่าของแอตทริบิวต์ ถ้าไม่มีแอตทริบิวต์ Atts มีค่าเป็นลิทซ์ว่าง
end(name)	</name>	
ref(Addr,Atts)	< a href="Addr" Atts>	Atts เหมือนกันกับใน begin

นอกจากนี้ยังมีคำสั่งอื่นๆ ของ PiLLow Library ที่ใช้งานในสคริปต์โปรแกรมอื่นต่อไปนี้

ตาราง 3.2 แสดงคำสั่งโพรลอคของ PiLLow ไบรารีในการสร้างเอกสาร HTML เพิ่มเติม

คำสั่งของ PiLLow ไบรารี	HTML ที่ได้	หมายเหตุ
comment(Comment)	<!-- Comment -->	
name(Text)	<name>Text</name>	
itemize(Items)	Data1 Data2 Data3 DataN 	Items คือลิสต์ของข้อมูลสมมุติให้ Items=[Data1,Data2,..,DataN]
//	 	
start_form	<form>	
end_form	</form>	

ในการทำงานโดยโหนด PiLLow Library ซึ่งเป็นฟรีแวร์เข้ามาใช้งานจะต้องมีการแก้ไขปรับปรุงไฟล์ pillow.pl บางส่วนเพื่อให้ทำงานกับ BinProlog ได้ โดยเพิ่มเพรคดิเคดที่จะทำการสร้างและใช้งานในโปรแกรมลงใน เพรคดิเคด module/2 เพื่อประกาศชื่อเพรคดิเคดก่อนการใช้งานจริง ในงานวิจัยนี้ได้เพิ่มไป 7 เพรคดิเคดคือ diary/3, invisible/2, reserved/1, can_scc/2, getenvstr/2, contains/2 และ date2day/2

เพรคดิเคด diary/3 เป็นข้อเท็จจริงที่ใช้เก็บข้อมูลการนัดหมายของไดอารี่ เพรคดิเคด invisible/2 เป็นกฎสำหรับการมองเห็นข้อมูลซึ่งใช้ในการกรอง diary/3 เพรคดิเคด reserved/1 เป็นกฎสำหรับการจองเวลาสวงนของไดอารี่เพื่อไม่ให้มีการนัดหมายเกิดขึ้น¹ เพรคดิเคดทั้ง 3 ตัวนี้เป็นเพรคดิเคดหลักของโปรแกรม

¹ดูโครงสร้างของเพรคดิเคดเหล่านี้เพิ่มเติมในหัวข้อ 4.2

สำหรับเพรตคิเคต `can_see/2` เป็นเพรตคิเคตระดับบนของเพรตคิเคต `invisible/2` เพรตคิเคต `getenvstr/2` สำหรับดึงชื่อของผู้ใช้งานโปรแกรมที่อยู่ทางด้านไคล่เอ็นท์ในขณะนั้นซึ่งจะถูกเก็บค่าไว้ใน "REMOTEUSER" เพรตคิเคต `contains/2` เป็นเพรตคิเคตตรวจสอบการมีอักขระภายใต้ชุดข้อมูลหนึ่ง เพรตคิเคต `date2day/2` เป็นเพรตคิเคตสำหรับแปลงวันที่เป็นวันในสัปดาห์เพรตคิเคตในส่วนนี้เป็นเพรตคิเคตที่โปรแกรมสร้างขึ้นเพื่อใช้งานกับการสร้างคำถามและตอบคำถาม ดังนั้นถ้าจะมีการเพิ่มเติมเพรตคิเคตอื่นๆ ในการถามคำถามจะต้องมีการประกาศเพิ่มเติมก่อนเสมอ โปรแกรมจึงจะสามารถทำงานได้

บทที่ 4

ลอจิกโคอาร์บินเว็บ

ลอจิกโคอาร์บินเว็บ คือ โปรแกรมโคอาร์เชิงตรรกที่ทำงานบนระบบเครือข่ายไบนามุม¹ กับ CGI Script² ที่เขียนโดย BinProlog³ และ Pillow Library⁴ โดยมีการซ่อนภาษาเชิงตรรกไว้เบื้องหลังการทำงานของระบบเครือข่ายไบนามุม เพื่อช่วยผู้ใช้ในการจัดการโคอาร์ดังต่อไปนี้คือ การบันทึกนัดหมาย การลบนัดหมาย การแก้ไขรายละเอียดการนัดหมาย การดูนัดหมาย และการสอบถามรายละเอียดการนัดหมาย โดยการทำงานอยู่ภายใต้เงื่อนไขเชิงตรรกที่ผู้ใช้กำหนดขึ้นให้กับโคอาร์ของตนเอง

การทำงานของลอจิกโคอาร์บินเว็บทำให้ผู้ใช้สะดวก และใช้งานได้ง่าย ผู้ใช้ไม่จำเป็นต้องมีความรู้ในเรื่องของภาษาไพโรลอค อย่างลึกซึ้งก็สามารถใช้งานได้ดี เพราะฟอร์มของ HTML จะเป็นอินเตอร์เฟซให้ระหว่างผู้ใช้กับ BinProlog นอกจากนี้ระบบเครือข่ายไบนามุมมีคุณสมบัติทางด้านระบบความปลอดภัยที่โปรแกรมลอจิกโคอาร์ต้องการใช้งานในความปลอดภัยของโคอาร์ได้ด้วย

การประมวลผลโดยใช้ภาษาไพโรลอคซึ่งในการวิจัยครั้งนี้ใช้ BinProlog ทำให้สามารถทำงานกับฐานข้อมูลแบบอนุमान, การวินิจฉัย (inference) เพื่อประมวลผลข้อมูลในเชิงตรรก, การแทนค่าข้อมูล (unification/pattern matching) การย้อนรอยเพื่อค้นหาคำตอบ (backtracking search) การตอบคำถามที่ยากและซับซ้อนได้ง่าย และการเพิ่มกฎให้ทำงานตามความต้องการของผู้ใช้

4.1 การทำงานของระบบลอจิกโคอาร์บินเว็บ

ลักษณะการทำงานของระบบลอจิกโคอาร์บินเว็บแสดงดังภาพประกอบ 4.1 เป็นลักษณะของสถาปัตยกรรมไคลเอ็นท์/เซิร์ฟเวอร์ (Client/Server)

4.1.1 ด้านเว็บเซิร์ฟเวอร์ ประกอบด้วย

4.1.1.1 สคริปต์ CGI ที่เขียนขึ้นโดย BinProlog สำหรับการทำงานหลัก 6 อย่างคือ

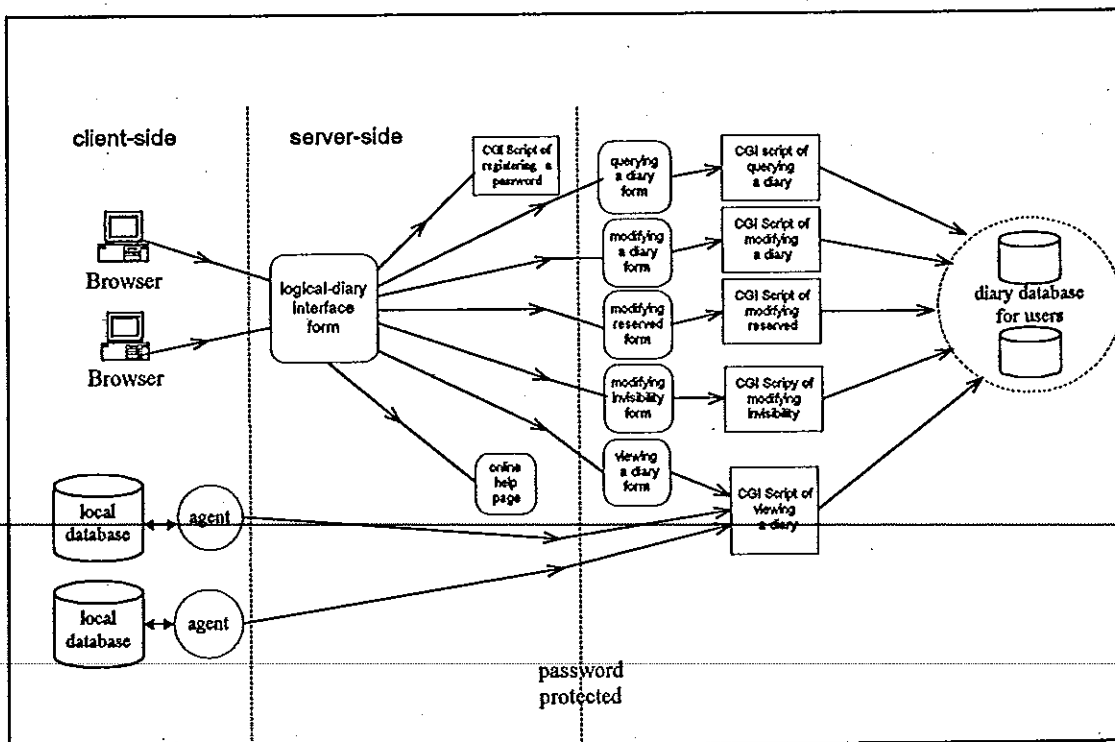
¹ ดูเครือข่ายไบนามุมในหัวข้อ 3.2

² ดู CGI Script ในหัวข้อ 3.2.2

³ ดู BinProlog ในหัวข้อ 3.3

⁴ ดู Pillow Library ในหัวข้อ 3.4

- การสมัครเป็นสมาชิกของระบบ (registering a password) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ cryptc.c , m_passwd.cgi, m_passwd.c และ p_passwd.pl
- การสอบถามข้อมูลของไดอารี่ (querying a diary) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ m_query.cgi, m_query.pl และ p_query.pl
- การเปลี่ยนแปลงข้อมูลในไดอารี่ (modifying a diary) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ m_change.cgi, m_change.pl และ p_change.pl
- การสร้างและยกเลิกกฎการจองเวลา (modifying reserved details) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ m_reserved.cgi, m_reserved.pl และ p_reserved.pl
- การสร้างและยกเลิกกฎการมองเห็นข้อมูลในไดอารี่ (modifying invisibility details) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ m_invisibility.cgi, m_invisibility.pl และ p_invisibility.pl
- การแสดงปฏิทินการนัดหมายรายเดือน (viewing a diary) มีชื่อโปรแกรมในการทำงานดังต่อไปนี้ m_view.cgi, m_view.pl และ p_view.pl



ภาพประกอบ 4.1 แสดงระบบบล็อกไดอารี่บนเว็บ

4.1.1.2 ฐานข้อมูลไดอารี่

ฐานข้อมูลไดอารี่เป็นฐานข้อมูลแบบอนุमान สำหรับเก็บข้อมูลการนัดหมายและกฎของผู้ใช้ โดยผู้ใช้แต่ละคนภายหลังจากการลงทะเบียนระบบจะสร้างเพิ่มข้อมูลสำหรับเก็บข้อมูลการนัดหมายอยู่ภายใต้เพิ่มข้อมูลที่ชื่อเหมือนกับชื่อผู้ใช้ในการเข้าใช้ระบบนามสกุล "db" และกฎอยู่ภายใต้เพิ่มข้อมูลนามสกุล "clause" การแยกการจัดเก็บเป็น 2 เพิ่มข้อมูลทำให้การทำงานสำหรับการเพิ่มและลบข้อมูลง่ายและสะดวกยิ่งขึ้น

ตัวอย่างของเพิ่มข้อมูล สมมุติชื่อผู้ใช้ในการเข้าใช้ระบบคือ "John" ดังนั้นจะมีเพิ่มข้อมูล 2 เพิ่มคือ "John.db" แสดงตัวอย่างข้อมูลในเพิ่มดังภาพประกอบ 4.2 และ "John.clause" แสดงตัวอย่างข้อมูลดังภาพประกอบ 4.3 และชื่อ ไดอารี่คือ John

```
diary('Board of directors', time(9,10,date(10,8,1998)),
item(topic('ordinary meeting',
'I must report on the progressional MML project'),
'M.125 ',8)). %ข้อเท็จจริงที่ 1

diary('#Jenny Jack',time(15,16,date(6,8,1998)),
item(topic('MML Project',
'Jenny and Jack report on the progressional MML project'),
'L.141 Laboratory',7)). %ข้อเท็จจริงที่ 2

diary('#Jenny Jack',time(15,16,date(7,8,1998)),
item(topic('MML Project',
'Jenny and Jack report on the progressional MML project'),
'L.141 Laboratory',7)). %ข้อเท็จจริงที่ 3

diary('#Jenny',time(20,21,date(15,8,1998)),
item(topic('Dinner',
'Jenny's Birthday'),
'VL. Restaurant',7)). %ข้อเท็จจริงที่ 4

diary('#Jenny',time(20,22,date(22,8,1998)),
item(topic('Movie',
'link("http://pangha.cs.psu.ac.th/~amrudee/WBLD/
movies/awfbigad.mov","See")'),
'Diana theatre',5)). %ข้อเท็จจริงที่ 5
```

ภาพประกอบ 4.2 แสดงเพิ่มข้อมูล John.db


```

(invisible(all,time(9,10,date(10,8,1998))))).      %กฎที่ 1
(invisible('Jack',time(_x13110,_x13111,_x13246)) :-
  date2day(_x13246,_x13518),
  (_x13518 = sat ; _x13518 = sun)).      %กฎที่ 2
(reserved(time(1,24,date(16,8,1998))))).      %กฎที่ 3
(reserved(time(_x9574,_x9611,_x9462)) :-
  _x9574 >= 19,_x9611 =< 24).      %กฎที่ 4

```

ภาพประกอบ 4.3 แสดงเพิ่มข้อมูล John.clause

ระบบความปลอดภัยของเครือข่ายอินเทอร์เน็ตช่วยให้ข้อมูลมีความปลอดภัยจากผู้ที่ไม่มีความรู้ที่เป็นเจ้าของไดอารี่เท่านั้นที่มีสิทธิ์ทำงานกับไดอารี่ของตนเอง ส่วนผู้ใช้คนอื่นๆ ก็ทำงานภายใต้กฎเกณฑ์ที่เจ้าของไดอารี่กำหนดขึ้นซึ่งได้แก่การมองเห็นข้อมูลภายในไดอารี่ การจองเวลาสงวน ผู้ที่ได้สมัครเป็นสมาชิกของระบบเท่านั้นมีสิทธิ์ในการเข้าถึงข้อมูล ซึ่งเป็นความปลอดภัยสำหรับข้อมูลในระดับแรกที่โปรแกรมจัดการให้

4.1.1.3 ฟอรัม HTML 6 ฟอรัม

ฟอรัมสำหรับผู้ใช้ในการทำงานในระดับบนสุดเพื่อติดต่อกับสคริปต์ CGI ทั้ง 6 ข้างต้น ซึ่งแต่ละฟอรัมจะมีลิงค์เชื่อมโยงถึงกันในรูปแบบของเมนู

ฟอรัมแรกของการเข้าสู่โปรแกรมคือภาพประกอบ 2.1 เมื่อผู้ใช้เข้าสู่ระบบโดยการคลิกที่ Member Entry จะเข้าสู่เมนูการทำงานของระบบ (ดูภาพของฟอรัมต่างๆ ในภาคผนวก ก)

- ฟอรัมการสอบถาม ภาพประกอบ ก.1(query.htm)
- ฟอรัมการเปลี่ยนแปลง diary/3 ภาพประกอบ ก.2 (change.htm)
- ฟอรัมการกำหนดและยกเลิกกฎ invisible/2 ภาพประกอบ ก.3 (invisible.htm)
- ฟอรัมการกำหนดและยกเลิกกฎ reserved/1 ภาพประกอบ ก.4 (reserved.htm)
- ฟอรัมการแสดงปฏิทินรายเดือน ภาพประกอบ ก.5(view.htm)
- เอกสารประกอบการใช้งาน (help.htm)

4.1.2 ด้านไคล์แอนท์

แบ่งผู้ใช้งานเป็น 2 กลุ่มคือ ผู้ใช้โปรแกรมที่ทำงานโดยผ่านโปรแกรมบราวเซอร์ และ โปรแกรม Logical Diary Agents on the Web

4.1.2.1 ผู้ใช้ที่ใช้งานผ่านโปรแกรมบราวเซอร์

ผู้ใช้ซึ่งอยู่ทางด้าน Client ใช้งานโปรแกรมโดยเรียกใช้โปรแกรมบราวเซอร์แล้วเข้าไปยัง URL ของ diary interface¹ ระบบจะสอบถาม username และ password ผู้ใช้ที่สมัครเป็นสมาชิกเท่านั้นที่มีสิทธิ์ในการเข้าไปยังฟอร์มต่างๆ เพื่อทำงานกับสคริปต์ CGI ยกเว้นฟอร์มแรกที่มีการสมัครเป็นสมาชิก ทุกๆ คนสามารถเข้าได้

โปรแกรมบราวเซอร์ที่ใช้ควรมีคุณสมบัติทำงานกับ JavaScript ได้ เช่น Netscape Gold ขึ้นไปหรือ Internet Explorer 4.0 ขึ้นไป เพราะในโปรแกรมมีการใช้ JavaScript ช่วยในการตรวจสอบการใส่ข้อมูลของผู้ใช้ให้ถูกต้องในระดับหนึ่งก่อนที่โปรแกรมบราวเซอร์จะทำการส่งข้อมูลทั้งหมดไปประมวลผลทางด้านเซิร์ฟเวอร์ เพื่อแบ่งเบาภาระการทำงานของโปรแกรม CGI Script ของทางด้านเซิร์ฟเวอร์ทำให้การทำงานเร็วขึ้น

4.1.2.2 โปรแกรมเอเจนต์โคออร์ดิเนตเว็บ (Logical Diary Agents on the Web)

โปรแกรม Logical Diary Agents on the web เป็นโปรแกรมผู้ช่วยสำหรับเฝ้าดูการเปลี่ยนแปลงของฐานข้อมูลโคออร์ดิเนตเว็บที่พัฒนาขึ้นโดยนางสาวเดือนเพ็ญ สัตถุศิริธรรม นักศึกษาระดับปริญญาโทสาขาวิทยาการคอมพิวเตอร์ในหัวข้อเรื่อง "Logical Diary Agents on the Web" ตัว Agent จะติดต่อเอาข้อมูลของโคออร์ดิเนตเว็บโดยผ่านทาง CGI Script ของการดูโคออร์ดิเนตเว็บ เพราะโปรแกรม CGI Script นอกเหนือจากการแสดงปฏิทินในรูปแบบกราฟิกรายเดือนของเอกสาร HTML แล้วยังซ่อนข้อมูลโคออร์ดิเนตเว็บที่อยู่ในรูปภาษาโปรแกรมไว้ภายใต้คอมเมนต์แท็ก (<!-- comment -->) และโปรแกรมแท็ก (<prolog> diary/3 </prolog>) ด้วยซึ่งจะไม่แสดงให้ผู้ใช้เห็นเป็นข้อความบนโปรแกรมบราวเซอร์ สำหรับให้โปรแกรม Agent นำไปสกัดเก็บไว้ใน Local Database เพื่อใช้ในการประมวลผลการทำงานต่อไป

ตัวอย่าง โค้ดเอกสาร HTML ที่ส่งกลับมาให้ Agent เมื่อ Agent ติดต่อกับ CGI-Script ในการดูโคออร์ดิเนตเว็บ (m_view.cgi)

¹ <http://pangba.cs.psu.ac.th/~amrudee/WBLD/>

```

<html>
<HEAD><TITLE>View diary</TITLE></HEAD>
<CENTER><H1><font color="#0000FF" size="5">John's
Diary</font></H1></CENTER><CENTER><H1>January
1998</H1></CENTER><CENTER>
<BODY BGCOLOR=white BACKGROUND="">
<TABLE BORDER=1 WIDTH=100%
CELLSPACING=1 CELLPADDING=1>
...
...
...
</TABLE>
<!--<prolog>
/* diary/3 facts ตัวอย่างข้อเท็จจริงที่ 1-5 */
</prolog-->
</html>

```

4.2 โครงสร้างฐานข้อมูลไดอารี่ (ดูภาพประกอบ 4.2 และ 4.3 ประกอบ)

จากภาพประกอบ 4.2 และ 4.3 ซึ่งแสดงรายละเอียดของแฟ้มข้อมูล John.db และ John.clause นั้นประกอบด้วย 3 เพรดคิเคตด้วยกันคือ คือ diary/3, invisible/2 และ reserved/1

4.2.1 ข้อเท็จจริง diary/3

diary/3 ใช้สำหรับเก็บข้อมูลการนัดหมายโดย 1 diary/3 แทน 1 การนัดหมาย มีรูปแบบโครงสร้างข้อมูลดังนี้

```
diary (Name, TimePeriod, Item) .
```

Name คือ ชื่อของบุคคล หรือ ชื่อไดอารี่ของผู้ใช้ในระบบที่เจ้าของไดอารี่ทำการนัดหมายด้วย ข้อเท็จจริงที่ 1 จากภาพประกอบ 4.2 เป็นชื่อของบุคคล แต่ถ้าเป็นกลุ่มชื่อของไดอารี่ในระบบในกรณีนัดหมายระหว่างหลายๆ ไดอารี่ ค่าของ Name นี้จะต้องขึ้นต้นด้วยสัญลักษณ์พิเศษ '#' นำหน้าชื่อ ไดอารี่ และแยกแต่ละชื่อ ไดอารี่ด้วยช่องว่าง เช่นข้อเท็จจริงที่ 2, 3, 4 และ 5 ซึ่งแต่ละข้อเท็จจริงจะหมายถึงเป็นการนัดหมายระหว่างไดอารี่ที่มีรายชื่อปรากฏทั้งหมด

TimePeriod คือ ช่วงเวลาและวันที่นัดหมาย มีโครงสร้างภายในดังนี้

```
time (StartSlot, EndSlot, date (Day, Month, Year))
```

StartSlot และ EndSlot คือเวลาเริ่มต้นและสิ้นสุดการนัดหมาย เป็นข้อมูลชนิดตัวเลขจำนวนเต็มที่มีค่าอยู่ในช่วงตั้งแต่ 1, 2, ..., 24

สำหรับ StartSlot ตัวเลขจำนวนเต็มใช้แทนเวลาเริ่มต้นของชั่วโมงดังนี้

1 แทน เพียงคืน,

2 แทน 1 นาฬิกา

3 แทน 2 นาฬิกา

...

24 แทน 23 นาฬิกา

สำหรับ EndSlot ตัวเลขจำนวนเต็มใช้แทนเวลาจบของชั่วโมงดังนี้

1 แทน 0:59 นาฬิกา

2 แทน 1:59 นาฬิกา

3 แทน 2 :59 นาฬิกา

...

24 แทน 23:59 นาฬิกา

การเปลี่ยนแปลงเวลาไปเป็นตัวเลขจำนวนเต็มทำได้ โดยการใช้ เพรตติเคต map_time_b/2(+,-) และ map_time_e/2(+,-) เป็นเพรตติเคต ที่ใช้ในการเปลี่ยนเวลาไปเป็นจำนวนเต็ม โดยมีทอมแรกเป็นเวลาจริง ทอมที่ 2 เป็นจำนวนเต็ม

ตัวอย่าง การทำงานของเพรตติเคต map_time_b/2 และ map_time_e/2

```
?-map_time_b('7.00', StartSlot).
StartSlot = 8
?-map_time_e('9.59', EndSlot).
EndSlot = 10
?-map_time_b(BTime, 9).
BTime = '8.00'
```

การใช้เลขจำนวนเต็มแทนเวลาของเวลาจริง ทำให้ง่ายต่อการทำงานของโปรแกรม และสามารถทำงานได้ซับซ้อนมากยิ่งขึ้น เช่นต้องการสอบถามหาค่าของนัดหมายที่ใช้เวลานัดหมายเพียง 2 ชั่วโมง ก็คือค้นหาค่าของ EndSlot - StartSlot = 1

date(Day, Month, Year) แทนวันเดือนปีที่ทำการนัดหมายโดย Day มีค่าอยู่ในช่วง 1-31 Month มีค่าอยู่ในช่วง 1-12 และ Year เป็นเลขจำนวนเต็ม 4 หลัก

Item เก็บรายละเอียดของการนัดหมาย มีโครงสร้างภายในดังนี้

```
item(topic(TopicTitle, Comments), Where, Importance)
```

TopicTitle, Comments และ Where เป็นข้อมูลชนิดสายอักขระโดยที่ TopicTitle เก็บหัวข้อการนัดหมาย Comments เก็บรายละเอียดพิเศษเพิ่มเติมสำหรับการนัดหมาย นอกจากนี้ข้อมูลในฟิลด์ Comment ยังสามารถกำหนดให้เชื่อมโยงไปยังเอกสารอื่นๆ บนเว็บ โดยมีรูปแบบของการเชื่อมดังนี้ link("URL", "TEXT") แสดงได้ดังข้อเท็จจริงที่ 5 Where เก็บสถานที่ทำการนัดหมาย Importance เป็นข้อมูลชนิดจำนวนเต็ม มีค่าอยู่ในช่วง 1-10 ใช้สำหรับกำหนดระดับความสำคัญของการนัดหมาย Importance ที่มีตัวเลขค่าสูงกว่าจะมีระดับความสำคัญสูงกว่าตัวเลขที่ต่ำกว่า

ความหมายของข้อเท็จจริงที่ 1 ดังแสดงในภาพประกอบ 4.2 John มีประชุมกับกรรมการบริหารในวันที่ 10 สิงหาคม 1998 ที่ห้อง M.125 เวลา 8.00-9.59 ซึ่ง John จะต้องเป็นผู้รายงานความก้าวหน้าของโครงการ MML ระดับความสำคัญของการประชุม John ให้เป็น 8

ความหมายของข้อเท็จจริงที่ 2 ดังแสดงในภาพประกอบ 4.2 John มีนัดระหว่างไคอารีกับ Jenny และ Jack ที่ห้องทดลอง L.141 เพื่อดูผลการทำงานของทั้งสองคนในโครงการ MML ในวันที่ 6 สิงหาคม 1998 เวลา 14.00-15.59 ระดับความสำคัญของการนัดหมาย John ให้เป็น 7

ความหมายของข้อเท็จจริงที่ 3 ดังแสดงในภาพประกอบ 4.2 John มีนัดระหว่างไคอารีกับ Jenny และ Jack ที่ห้องทดลอง L.141 เพื่อดูผลการทำงานของทั้งสองคนในโครงการ MML ในวันที่ 7 สิงหาคม 1998 เวลา 14.00-15.59 ระดับความสำคัญของการนัดหมาย John ให้เป็น 7

ความหมายของข้อเท็จจริงที่ 4 ดังแสดงในภาพประกอบ 4.2 John มีนัดระหว่างไคอารีเพื่อทานอาหารมื้อค่ำกับ Jenny ที่ภัตตาคาร VL. ในวันที่ 15 สิงหาคม 1998 เวลา 19.00-20.59 เนื่องในวันคล้ายวันเกิดของ Jenny ระดับความสำคัญของการนัดหมาย John ให้เป็น 7

ความหมายของข้อเท็จจริงที่ 5 ดังแสดงในภาพประกอบ 4.2 John มีนัดระหว่างไคอารีเพื่อดูภาพยนตร์กับ Jenny ที่โรงภาพยนตร์ Diana ในวันที่ 22 สิงหาคม 1998 เวลา 19.00-21.59 John ได้สร้างลิงค์ไปยังแฟ้มข้อมูลภาพยนตร์ตัวอย่างเพื่อให้ Jenny ได้ดูล่วงหน้าด้วย

4.2.2 กฎ invisible/2 (รูปภาพประกอบ 4.3)

invisible/2 ใช้สำหรับกรองข้อเท็จจริง diary/3 ที่เจ้าของไดอารี่ไม่ต้องการให้ผู้ใช้คนอื่น ๆ เห็นข้อมูลการนัดหมายของคุณ หรือคุณต้องการปกปิดข้อมูลการนัดหมายที่เป็นความลับสำหรับบางคน คุณสามารถกำหนดกฎเงื่อนไขการทำงานของไดอารี่ขึ้นได้ด้วยตัวคุณเองกฎ invisible/2 จะมีผลกับผู้ใช้ทุกคนยกเว้นเจ้าของไดอารี่ invisible/2 มีโครงสร้างของกฎ ดังนี้

```
invisible(UserID,TimePeriod):- /* extra conditions */
```

UserID เป็นชื่อของผู้ใช้ในระบบที่ผู้ใช้ระบุไม่ต้องการให้เห็นข้อมูล หรือใช้คำว่า "all" แทนผู้ใช้ทุกคนของระบบ แต่ยกเว้นเจ้าของไดอารี่ TimePeriod มีรูปแบบเหมือน TimePeriod ใน diary/3 เป็นช่วงวันเวลาของนัดหมายที่ไม่ต้องการให้แสดงข้อมูล extra conditions สำหรับเงื่อนไขของกฎ invisible/2 ซึ่งสามารถกำหนดให้มีความสัมพันธ์กับเพรดิคตอื่นๆ ที่ใช้งานได้

ความหมายของกฎที่ 1 ดังแสดงในภาพประกอบ 4.3 ห้ามทุกคนดูนัดหมายในวันที่ 10 สิงหาคม 1998 เวลา 8.00-9.59 ว่า John มีนัดหมายกับใครบ้าง

ความหมายของกฎที่ 2 ดังแสดงในภาพประกอบ 4.3 ห้าม Jack ดูทุกนัดหมายของ John ในทุกๆวันเสาร์และอาทิตย์

4.2.3 กฎ reserved/1 (รูปภาพประกอบ 4.3)

reserved/1 เป็นกฎที่ใช้สำหรับกำหนดช่วงเวลาที่ไม่ให้มีการนัดหมายเกิดขึ้นได้ภายในไดอารี่ของคุณ มีโครงสร้างของกฎ ดังนี้

```
reserved(TimePeriod): -/*extra conditions */
```

TimePeriod มีโครงสร้างเหมือนกับ TimePeriod ใน invisible/2 เป็นช่วงวันเวลาที่กำหนดไว้ว่าไม่อนุญาตให้มีการทำนัดหมายซึ่งกฎ นี้จะมีผลต่อทั้งเจ้าของไดอารี่และผู้ใช้คนอื่น ๆ ที่ต้องการจะทำนัดหมายกับไดอารี่ของคุณ

Extra conditions สำหรับเงื่อนไขของกฎ reserved/2 ซึ่งสามารถกำหนดให้มีความสัมพันธ์กับเพรดิคตอื่นๆ ที่ใช้งานได้

ความหมายของกฎที่ 3 ดังแสดงในภาพประกอบ 4.3 ห้ามมีการนัดหมายเกิดขึ้นในวันอาทิตย์ที่ 16 สิงหาคม 1998

ความหมายของกฎที่ 4 ดังแสดงในภาพประกอบ 4.3 ห้ามทำการนัดหมายในช่วงเวลา 18.00 -23.59 ของทุกวัน

4.3 การสอบถามฐานข้อมูลแบบอนุमानของไดอารี่

การสอบถามทำงานโดยผ่านระบบเครือข่ายใยแมงมุมที่ซ่อนภาษาเชิงตรรกไว้เบื้องหลังการทำงาน ผู้ใช้ทำงานได้โดยผ่านฟอร์มการสอบถาม (ภาพประกอบ ก.1) การสอบถามไดอารี่ คือ การสอบถามข้อมูลการนัดหมาย โดยการใส่ค่าของข้อมูลที่ต้องการจะสอบถามลงไปในฟอร์มเพื่อสร้างคำถาม ค่าข้อมูลใดที่ต้องการทราบคำตอบให้ใส่ตัวแปรไดอารี่¹ (ค่าที่ขึ้นต้นด้วย "dv") ส่วนค่าข้อมูลใดที่ไม่ต้องการทราบคำตอบก็เว้นว่างไว้ ผู้ใช้สามารถจะสร้างเงื่อนไขสำหรับคำถามนั้นๆ ได้ โดยใช้ร่วมกับตัวแปรไดอารี่ ทำให้ผู้ใช้สามารถถามคำถามที่ซับซ้อนได้ง่าย

4.3.1 การทำงานของโปรแกรมในการสอบถามข้อมูล

การทำงานของโปรแกรมในการสอบถามข้อมูลนั้นจะต้องแปลงคำถามของผู้ใช้ให้เป็นคำถามของ BinProlog โดยการเปลี่ยนตัวแปรไดอารี่และเทอมเงื่อนไขของผู้ใช้ ให้เป็นตัวแปรและเทอมของ BinProlog แล้วจึงสร้างคำถาม เพราะคติเคหลักในการทำงานเพื่อให้ได้คำตอบดังภาพประกอบ 4.4 ถึง 4.7 คือ `execute_query/12(+,+,+,+,+,+,+,+,+,+,+)`

เพราะคติเค `execute_query/12` จะรับค่าของอินพุตทั้งหมดของ `diary/3` ชื่อไดอารี่ที่ต้องการจะสอบถาม และเทอมเงื่อนไขในการสอบถามมาตรวจสอบและประมวลผลจนได้คำตอบที่สอดคล้องแล้วแสดงผลลัพธ์กลับไปในรูปแบบของ HTML

โค้ดการทำงานของ `execute_query/12`

¹ดูรายละเอียดเพิ่มเติมในหน้าที่ 54

```

execute_query(DName,Who,Day,Month,Year,BTm,ETm,Where,
              Topic,Comment,Importance,Condi):-
    form_empty_value(Condi),% Is Condition field empty?
    form_empty_value(DName),!, %Is diary name field empty?
    getenvstr('REMOTE_USER',UserStr),
    %get the name of remote user
    name(User,UserStr), % user
    make_query(3,User,Who,Day,Month,Year,BTm,ETm,
              Where,Topic,Comment,Importance,_,
              Goal,DvL,GL),
    % no conditions and ask your diary.
    execute_goal(Goal,GL,DvL).
execute_query(DName,Who,Day,Month,Year,BTm,ETm,Where,
              Topic,Comment,Importance,Condi):-
    form_empty_value(Condi),
    not form_empty_value(DName),user(DName,_,!),
    make_query(4,DName,Who,Day,Month,Year,
              BTm,ETm,Where,Topic,Comment,
              Importance,_,Goal,DvL,GL),% no condition and ask
    execute_goal(Goal,GL,DvL).% other diary
execute_query(DName,Who,Day,Month,Year,BTm,ETm,
              Where,Topic,Comment,Importance,Condi):-
    not form_empty_value(Condi),
    form_empty_value(DName),!,
    getenvstr('REMOTE_USER',UserStr),
    name(User,UserStr),
    make_query(1,User,Who,Day,Month,Year,
              BTm,ETm,Where,Topic,Comment,
              Importance,Condi,Goal,DvL,GL),
    % have conditions and ask your diary
    execute_goal(Goal,GL,DvL).
execute_query(DName,Who,Day,Month,Year,BTm,ETm,Where,
              Topic,Comment,Importance,Condi):-
    not form_empty_value(Condi),
    not form_empty_value(DName),user(DName,_,!),
    make_query(2,DName,Who,Day,Month,Year,
              BTm,ETm,Where,Topic,Comment,
              Importance,Condi,Goal,DvL,GL),
    % have conditions and ask other diary
    execute_goal(Goal,GL,DvL).
execute_query(DName,_,_,_,_,_,_,_,_,_,_):-
    write_error_html(DName).
    % no diary name -> write error message

```

form_empty_value/1 เป็นเพรคดิเคตสำหรับตรวจสอบค่าของอินพุตที่รับมาจากฟอร์ม ถ้าไม่มีค่าข้อมูลเพรคดิเคตนี้ จะทำงานสำเร็จ แต่ถ้ามีค่าข้อมูลเพรคดิเคตนี้ จะไม่สำเร็จ

make_query/16 (+,+,+,+,+,+,+,+,+,+,+,+,-,-) ทำหน้าที่สร้างคำถามในรูปของภาษาโพรลอค ผลลัพธ์ที่ได้คือคำถามภาษาโพรลอค (Goal) ลิสต์ตัวแปรของระบบที่ใช้แทนตัวแปรไดอารี่แต่ละคู่ที่สัมพันธ์กัน (DvL) ภายใต้อุปกรณ์เพรคดิเคต dv(BinPrologTerm, DiaryVariable) และลิสต์ของตัวแปรระบบ (GL) เพรคดิเคตสำคัญในการใช้งานคือ build_diary0/5(+,+,+,-,-) สำหรับสร้าง diary/3 และ dv_term/4 (+,-,+,-) สำหรับเปลี่ยนเทอมเงื่อนไขของผู้ใช้เป็นเทอมของระบบ

Web Base Logical Diary - Netscape

http://pangha.ct.psu.ac.th/~arrudoe/WBLD/diary.html/DiaryFrame.htm

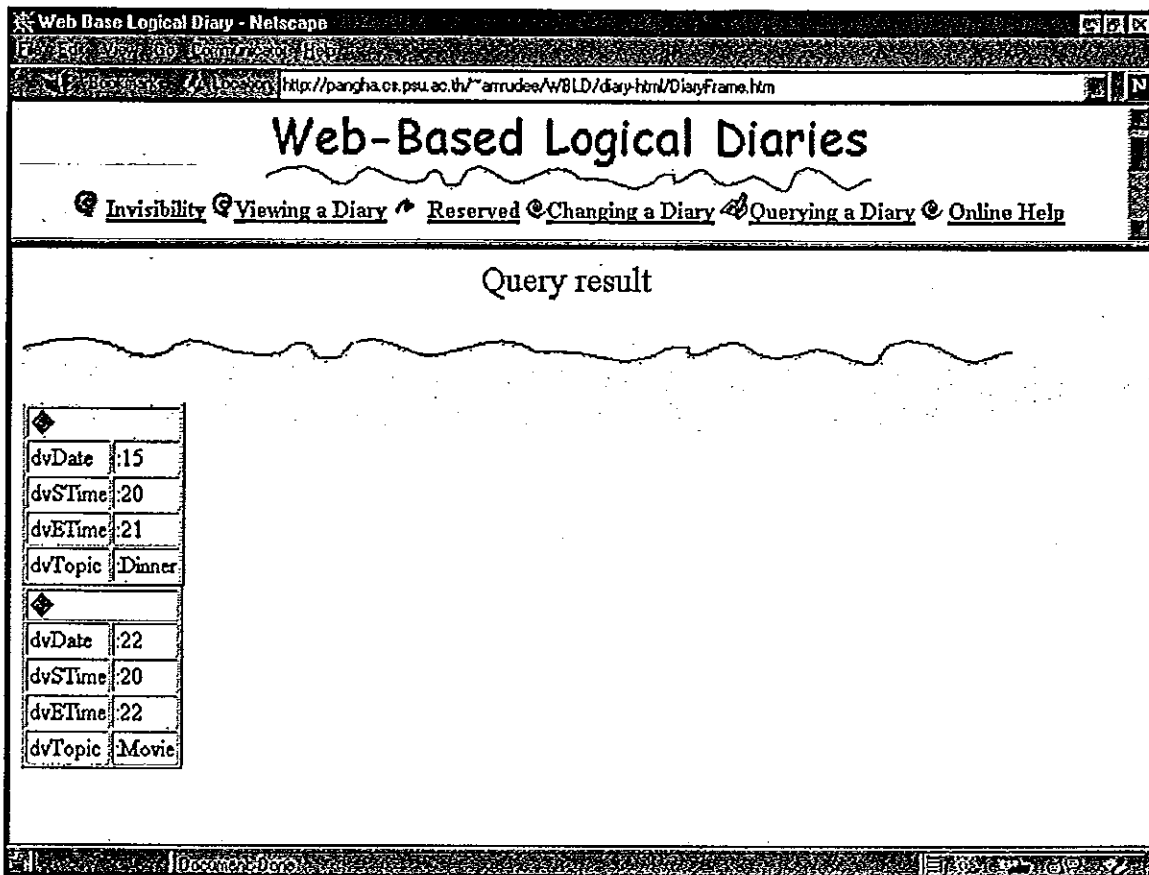
Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Querying a Diary

DiaryName:	<input type="text"/>	eg. Jack, Jenny, John or space for your diary
Who	<input type="text" value="#Jenny"/>	eg. Jane, Jim, #Jenny Jack or diary variable
DatePeriod:	<input type="text" value="dvDate"/> / <input type="text" value="8"/> // <input type="text" value="1998"/>	integer 1-31/1-12/1998 or diary variable
TimePeriod:	<input type="text" value="dvSTime"/> to <input type="text" value="dvETime"/>	integer 1-24 (more details) or diary variable
Where	<input type="text"/>	string or diary variable
Topic	<input type="text" value="dvTopic"/>	string or diary variable
Comment	<input type="text"/>	string or diary variable
Importance:	<input type="text"/>	integer 1-10 or diary variable
Conditions	<div style="border: 1px solid black; height: 40px; width: 100%;"></div> <p style="font-size: small;">Don't press enter for new line. See examples</p>	
<input type="button" value="EXECUTE"/> <input type="button" value="CLEAR"/>		

ภาพประกอบ 4.4.ก แสดงหน้าจอการสอบถามนัดหมายกรณีไม่มีเงื่อนไขของคำถาม
 คำถาม : John ถามไต่สารีของตนเองมีนัดหมายกับไต่สารีของ Jenny ในเดือน สิงหาคม 1998 ในวัน
 และเวลาใด เรื่องอะไรบ้าง



ภาพประกอบ 4.4.ข แสดงคำตอบของคำถามในภาพประกอบ 4.4.ก

คำตอบ : มี 2 คำตอบคือ John มีนัดกับ Jenny ในวันที่ 15 เวลา 18.00-19.59 เพื่อทานอาหารค่ำ และ ในวันที่ 22 เวลา 19.00-21.59 เพื่อดูภาพยนตร์ ซึ่งคำตอบจะตอบภายใต้ชื่อตัวแปรแต่ละตัวที่ผู้ใช้กำหนด

Web Base Logical Diary - Netscape

http://pangha.or.psu.ac.th/~annudee/WBLD/diary.html/DiaryFrame.htm

Web-Based Logical Diaries

Invisibility
 Viewing a Diary
 Reserved
 Changing a Diary
 Querying a Diary
 Online Help

Querying a Diary

DiaryName:	<input type="text"/>	eg. Jack, Jenny, John or space for your diary
Who	<input type="text" value="dvWho"/>	eg. Jane, Jim, #Jenny Jack or diary variable
DatePeriod:	<input type="text"/> / <input type="text"/> / <input type="text"/>	integer 1-31/1-12/1998 or diary variable
TimePeriod:	<input type="text"/> to <input type="text"/>	integer 1-24 (more details) or diary variable
Where	<input type="text" value="dvWhere"/>	string or diary variable
Topic	<input type="text" value="dvTopic"/>	string or diary variable
Comment	<input type="text" value="dvComment"/>	string or diary variable
Importance:	<input type="text"/>	integer 1-10 or diary variable
Conditions	<input ("mml",="" contains="" dvcomment))"="" dvtopic);="" mml",="" type="text" value="(contains ("/>	
	Don't press enter for new line. See examples	
	<input type="button" value="EXECUTE"/> <input type="button" value="CLEAR"/>	

Documents: []

ภาพประกอบ 4.5.ก แสดงหน้าจอการสอบถามกรณีมีเงื่อนไขของคำถาม

คำถาม : John มีนัดหมายใดบ้างที่เกี่ยวข้องกับ MML Project โดยต้องการคำตอบเฉพาะ กับใคร ที่ไหน หัวข้อเรื่องอะไรบ้าง

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~amrudee/WBLD/day.html/DiaryFrame.htm

Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Query result

dvWho	Board of directors
dvWhere	M.125
dvTopic	ordinary meeting
dvComment	I must report the progressional MML project
dvWho	Jenny Jack
dvWhere	L.141 Laboratory
dvTopic	MML Project
dvComment	Jenny and Jack report progressional MML project
dvWho	Jenny Jack
dvWhere	L.141 Laboratory
dvTopic	MML Project
dvComment	Jenny and Jack report progressional MML project

ภาพประกอบ 4.5.ข แสดงคำตอบของคำถามในภาพประกอบ 4.5.ก

คำตอบ : มี 3 คำตอบ (ข้อเท็จจริงที่ 1, 2 และ 3) คือทุกนัดหมายที่มีข้อความ "MML" อยู่ใน Topic และ Comment ฟิลด์

ตัวอย่าง การสร้างคำถามของไพลอตของภาพประกอบ 4.4.ก ข้อมูลที่มีลักษณะ _x20955 นี้หมายถึงข้อมูลชนิดตัวแปรในระบบไพลอตที่ยังไม่ได้ถูกให้ค่า

```
?- make_query(3, 'John', '#Jenny', 'dvDate', 8, 1998,
  'dvSTime', 'dvETime', '$empty', 'dvTopic',
  '$empty', '$empty', '$empty',
  Query, DvL, GL).
Query=findall(ans([_x21103, _x21040, _x21001,
  _x20955, _x20950, _x20960, _x20970]),
  diary('#Jenny', time(_x21040, _x21001,
  date(_x21103, 8, 1998)),
  item(topic(_x20950, _x20955),
  _x20960, _x20970)), _x3483),
DvL=[dv(_x21103, dvDate), dv(_x21040, dvSTime),
  dv(_x21001, dvETime), dv(_x20955, dv1),
  dv(_x20950, dvTopic), dv(_x20960, dv2),
  dv(_x20970, dv3)],
GL=_x3483
```

ตัวอย่าง การสร้างคำถามของโพรลอคของภาพประกอบ 4.5.ก

```
?- make_query(1, 'John', dvWho, '$empty', '$empty', '$empty',
'$empty', '$empty', 'dvWhere', 'dvTopic',
'dvComment', '$empty',
'(contains("MML", dvTopic);
contains("MML", dvComment))',
Query, DvL, GL).
Query=findall(ans([_x21236, _x21342, _x21343,
_x21344, _x21345, _x21346,
_x21419, _x21380, _x21341, _x21347]),
(diary(_x21236, time(_x21342, _x21343,
date(_x21344, _x21345, _x21346)),
item(topic(_x21380, _x21341), _x21419, _x21347)),
(contains([77,77,76], _x21380) ;
contains([77,77,76], _x21341))), _x3781),
DvL=[dv(_x21236, dvWho), dv(_x21342, dv1),
dv(_x21343, dv2), dv(_x21344, dv3),
dv(_x21345, dv4), dv(_x21346, dv5),
dv(_x21419, dvWhere), dv(_x21380, dvTopic),
dv(_x21341, dvComment), dv(_x21347, dv6)],
GL=_x3781
```

การสร้างคำถามในแต่ละครั้งนั้นจะแตกต่างกัน ขึ้นอยู่กับสอบถามไคอารี่ของตนเองหรือของผู้อื่นและมีเงื่อนไขเพิ่มเติมในการสอบถามหรือไม่ เพรตติเคดข้างล่างนี้สำหรับสร้างคำถาม 4 กรณี

make_query1/16(+,+,+,+,+,+,+,+,+,+,+,+,+,+,+,+)

สำหรับเจ้าของไคอารี่ และมีเงื่อนไข

make_query2/16(+,+,+,+,+,+,+,+,+,+,+,+,+,+,+,+)

สำหรับคนที่ไม่ใช่เจ้าของไคอารี่ และมีเงื่อนไข

make_query3 /15(+,+,+,+,+,+,+,+,+,+,+,+,+,+,+)

สำหรับเจ้าของไคอารี่ และ ไม่มีเงื่อนไข

make_query4 /15(+,+,+,+,+,+,+,+,+,+,+,+,+,+,+)

สำหรับคนที่ไม่ใช่เจ้าของไคอารี่ และ ไม่มีเงื่อนไข

โค้ดของ make_query/16 มีดังนี้

```

make_query(1, DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7,
           Arg8, Arg9, Arg10, Condi, Query, DvL, GL) :-
    make_query1(DName, Arg1, Arg2, Arg3, Arg4, Arg5,
               Arg6, Arg7, Arg8, Arg9, Arg10, Condi, Query, DvL, GL).
make_query(2, DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7,
           Arg8, Arg9, Arg10, Condi, Query, DvL, GL) :-
    make_query2(DName, Arg1, Arg2, Arg3, Arg4, Arg5,
               Arg6, Arg7, Arg8, Arg9, Arg10, Condi, Query, DvL, GL).
make_query(3, DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7,
           Arg8, Arg9, Arg10, _, Query, DvL, GL) :-
    make_query3(DName, Arg1, Arg2, Arg3, Arg4, Arg5,
               Arg6, Arg7, Arg8, Arg9, Arg10, Query, DvL, GL).
make_query(4, DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6, Arg7,
           Arg8, Arg9, Arg10, _, Query, DvL, GL) :-
    make_query4(DName, Arg1, Arg2, Arg3, Arg4, Arg5,
               Arg6, Arg7, Arg8, Arg9, Arg10, Query, DvL, GL).

make_query1(DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6,
            Arg7, Arg8, Arg9, Arg10, Condi,
            findall(ans(VarL), (Diary, Condi1), GL), DvL, GL) :-
    build_diary0(DName, [Arg10, Arg9, Arg8, Arg7, Arg6, Arg5,
                        Arg4, Arg3, Arg2, Arg1], Diary, DvL, VarL),
    name(Condi, C1), atom_chars(C2, C1),
    dv_term(C2, Condi1, DvL, _).

make_query2(DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6,
            Arg7, Arg8, Arg9, Arg10, Condi,
            findall(ans(VarL), (diary(Who, Time, Item), Condi1,
                                   can_see(User, Time)), GL), DvL, GL) :-
    getenvstr('REMOTE_USER', UserStr),
    name(User, UserStr),
    build_diary0(DName, [Arg10, Arg9, Arg8, Arg7, Arg6, Arg5,
                        Arg4, Arg3, Arg2, Arg1], diary(Who, Time, Item),
                DvL, VarL),
    name(Condi, C1), atom_chars(C2, C1),
    dv_term(C2, Condi1, DvL, _).

make_query3(DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6,
            Arg7, Arg8, Arg9, Arg10,
            findall(ans(VarL), Diary, GL), DvL, GL) :-
    build_diary0(DName, [Arg10, Arg9, Arg8, Arg7, Arg6, Arg5,
                        Arg4, Arg3, Arg2, Arg1], Diary, DvL, VarL).

make_query4(DName, Arg1, Arg2, Arg3, Arg4, Arg5, Arg6,
            Arg7, Arg8, Arg9, Arg10,
            findall(ans(VarL),
                    (diary(Who, Time, Item), can_see(User, Time)), GL),
            DvL, GL) :-
    getenvstr('REMOTE_USER', UserStr), name(User, UserStr),
    build_diary0(DName, [Arg10, Arg9, Arg8, Arg7, Arg6, Arg5,
                        Arg4, Arg3, Arg2, Arg1], diary(Who, Time, Item),
                DvL, VarL).

```

เพรตคิเคต `can_see/2` ใช้สำหรับตรวจสอบความสัมพันธ์กับ `invisible/2` clause ว่าผู้ใช้คนที่กำลังสอบถามอยู่มีสิทธิ์ในการดูข้อมูลในช่วงวันเวลาที่ระบุหรือไม่ จะไม่เรียกใช้เพรตคิเคตนี้กรณีเป็นเจ้าของโคอาร์ดูข้อมูลในโคอาร์ดูของตัวเอง

```
can_see(User,Time):-
    not invisible(User,Time),not invisible(all,Time).
```

`execte_goal/3(+,-,-)` ทำหน้าที่ประมวลผลคำสั่งเป้าหมายที่ได้จากเพรตคิเคต `make_query` โดยเรียกใช้เพรตคิเคต `call/1(+)` และส่งผลลัพธ์ที่ได้ไปเขียนโดยใช้เพรตคิเคต `write_ansL/3(+,+,-)` เพื่อเลือกเอาคำตอบเฉพาะเทอมที่ต้องการ เขียนในรูปของตารางโดยใช้ภาษา HTML ที่ต้องการโดยพิจารณาเฉพาะตัวแปรที่มีความสัมพันธ์กับตัวแปรโคอาร์ดูเท่านั้น หลังจากนั้นนำตารางคำตอบเหล่านั้นส่งกลับไปแสดงให้ผู้ใช้ทราบ โดยเพรตคิเคต `write_output_html/1(+)`

`execute_goal/3` มีโค้ดการทำงานดังนี้

```
execute_goal(Goal, GL, DvL):-
    call(Goal),
    write_ansL(GL, DvL, Ans),
    write_output_html(Ans).

write_ansL([],_, ' ').% write answer of the goal
% select write only diary variable
write_ansL([ans(L)|Gls], DvL, TBAns):-
    write_ansL(Gls, DvL, AnsL),
    write_ans(L, DvL, Ans),
    open_table(OT),close_table(CT),
    build_tag2([OT,Ans,AnsL,CT],TBAns).

% write the answer in table of html format
write_ans([],_, '</table>').
write_ans([V|Vs], [dv(_,Name)|DvL], AnsL):-
    write_ans(Vs, DvL, AnsL1),
    map_value(V, V1),chk_link(V1, V2),
    nskip_name(Name),
    write_row(Name, V2, W_Row),
    build_tag2([W_Row,AnsL1],AnsL).
write_ans([_|Vs], [_|DvL], AnsL):-
    write_ans(Vs, DvL, AnsL).
```

4.3.2 การสร้าง diary/3

ขั้นตอนการสร้าง diary/3 นั้นเป็นขั้นตอนที่ซับซ้อนและสำคัญเพราะเกือบทุกๆ โมดูลจะต้องทำงานในส่วนนี้ ดังนั้นผู้เขียนจึงขอแยกออกมาอธิบายเป็นอีกหัวข้อหนึ่งเพื่อจะได้อ้างอิงถึงในภายหลัง

การสร้าง diary/3 เพื่อใช้ในการสร้างคำถามภาษาโปรแกรมและการประมวลผลกับข้อมูลนั้นเรียกใช้เพรตคิเคต build_diary0/5(+,+,,-,-) ซึ่งจะเป็นเพรตคิเคตระดับบนสุดในการทำงาน การทำงานเริ่มจากเปิดฐานข้อมูลที่ต้องการ แล้วโหลดข้อมูลทั้งหมดลงบนหน่วยความจำ ก่อนที่จะเรียก build_diary/4(+,,-,-) เพื่อสร้าง diary/3

ตัวอย่าง การทำงานของเพรตคิเคต build_diary/4 โดยใช้ข้อมูลจากภาพประกอบ 4.4.ก

```
?-build_diary('John', [ '#Jenny',
  'dvDate', 8, 1998, 'dvSTime', 'dvETime', '$empty',
  'dvTopic', '$empty', '$empty', '$empty'], Diary, DvL, VarL).

Diary=diary('#Jenny', time(_x21040, _x21001,
  date(_x21103, 8, 1998)),
  item(topic(_x20950, _x20955),
    _x20960, _x20970)), _x3483),
DvL=[dv(_x21103, dvDate), dv(_x21040, dvSTime),
  dv(_x21001, dvETime), dv(_x20955, dv1),
  dv(_x20950, dvTopic), dv(_x20960, dv2),
  dv(_x20970, dv3)],
VarL=[_x21103, _x21040, _x21001,
  _x20955, _x20950, _x20960, _x20970]
```

ตัวอย่าง การทำงานของเพรตคิเคต build_diary/4 ใช้ข้อมูลจากภาพประกอบ 4.5.ก

```
?-build diary('John', [ 1, 'John', dvWho, '$empty',
  '$empty', '$empty', '$empty', '$empty', 'dvWhere',
  'dvTopic', 'dvComment', '$empty',
  '(contains("MML", dvT); contains("MML", dvC))'],
  Diary, DvL, VarL).

Diary=diary(_x21236, time(_x21342, _x21343,
  date(_x21344, _x21345, _x21346)),
  item(topic(_x21380, _x21341), _x21419, _x21347)),
  (contains([77, 77, 76], _x21380) ;
  contains([77, 77, 76], _x21341))), _x3781),
DvL=[dv(_x21236, dvWho), dv(_x21342, dv1),
  dv(_x21343, dv2), dv(_x21344, dv3),
  dv(_x21345, dv4), dv(_x21346, dv5),
  dv(_x21419, dvWhere), dv(_x21380, dvTopic),
  dv(_x21341, dvComment), dv(_x21347, dv6)],
VarL=[_x21236, _x21342, _x21343, _x21344, _x21345,
  _x21346, _x21419, _x21380, _x21341, _x21347]
```


โค้ดการทำงานของ build_diary0/5

```

build_diary0(DN,Args,Diary,DvL,VarL):-
    add_ext(DN,'.db',DName),% load database into memory
    add_ext(DN,'.clause',CName),
    load_diary(DName),
    load_clause(CName),
    % build diary/3
    build_diary(Args,Diary,DvL,VarL).

build_diary([Arg10,Arg9,Arg8,Arg7,Arg6,Arg5,Arg4,
            Arg3,Arg2,Arg1],diary(Who,Time,Item),DvL,VarL):-
    %change empty string to diary variable (dv1,dv2,..)
    conv_str([Arg10,Arg9,Arg8,Arg7,Arg6,Arg5,
            Arg4,Arg3,Arg2,Arg1],ArgL),
    %check diary variable and change to system variable
    check_var0(ArgL,VarL,DvL,PArgL),
    get_names([1],PArgL,Arg),
    get_vars(DvL,Arg,[Who]),
    build_date(PArgL,DvL,Date),% build date functor
    build_time(PArgL,Date,DvL,Time),% build time functor
    build_topic(PArgL,DvL,Topic),% build topic functor
    build_item(PArgL,Topic,DvL,Item).%build item functor

```

การทำงานของ build_diary/4 เริ่มต้นด้วยการเรียกใช้ conv_str/2(+,-) เพื่อแปลงเทอมอินพุตซึ่งเป็น empty string ให้มีค่าเป็นตัวแปรโคออร์ดิเนตที่ขึ้นต้นด้วย "dv" แล้วตามด้วยเลขจำนวนเต็มเช่น dv1 และ dv2 เพราะโปรแกรมจะได้แยกออกว่าเป็นตัวแปรที่ผู้ใช้ไม่สนใจเพื่อว่าเมื่อแสดงผลลัพธ์จะไม่แสดงผลของตัวแปรที่มีชื่อ dv1, dv2, ... ส่วนเทอมอินพุตตัวอื่นให้คงค่าไว้เหมือนเดิม

เหตุผลที่มีตัวแปรโคออร์ดิเนตในการทำงาน เพราะใน BinProlog ไม่สามารถทราบชื่อของตัวแปรได้ และถ้าใช้อักขระตัวใหญ่ขึ้นต้นจะหมายถึงตัวแปร ดังนั้นการใช้อักขระตัวเล็กทำให้สะดวกในการพิจารณาว่าไม่เป็นตัวแปรโคออร์ดิเนตหรือเป็นตัวแปรโคออร์ดิเนต และการใช้ตัวแปรโคออร์ดิเนตทำให้สามารถทราบถึงชื่อของตัวแปรได้ด้วย

ต่อมาทำการตรวจสอบตัวแปรโคออร์ดิเนตโดยเพรตดิเคต check_var0/3(+,-,-). เพื่อสร้างตัวแปร BinProlog สำหรับตัวแปรโคออร์ดิเนตแต่ละตัว ค่าที่ส่งกลับไปที่ชื่อชุดของตัวแปรทั้งหมด (VarL) คู่ลำดับความสัมพันธ์ระหว่างตัวแปรโคออร์ดิเนตและตัวแปรของ BinProlog (DvL) และ ชื่อของตัวแปรทั้งหมดเรียงลำดับสลับกับ ArgL ใหม่ (PArgL) จากนั้นก็สร้าง diary/3 จากเทอมของ BinProlog โดยเพรตดิเคต build_date/3(+,+,-), build_time/4(+,+,+,-),build_topic/3(+,+,-), build_item/4(+,+,+,-).

ตัวอย่าง การใช้งาน เปรตดิเคด check_var/0 โดยใช้ข้อมูลจากภาพประกอบ 4.4.ก

```
?-check_var0([ '#Jenny','dvDate',8,1998,'dvSTime','dvETime',
               '$empty','dvTopic','$empty','$empty','$empty'],
             VarL,DvL,PArgL).
VarL=[_x21103,_x21040,_x21001,_x20955,_x20950,
       _x20960,_x20970]
DvL=[dv(_x21103,dvD),dv(_x21040,dvS),
      dv(_x21001,dvE),dv(_x20955,dv1),
      dv(_x20950,dvT),dv(_x20960,dv2),
      dv(_x20970,dv3)],
PArgL=['dv3','dv2','dvTopic','dv1','dvETime','dvSTime',
       '1998','8','dvDate','#Jenny']
```

โค้ดการทำงานของ check_var/4 มีดังนี้

```
check_var0(Args,VarL,DvL,PArgL):-
    check_vars(Args,[],VarL,[],DvL,[],PArgL).

check_vars([],VarL,VarL,DvL,DvL,ArgL,ArgL).
check_vars([Arg|Args],VarL,NVarL,DvL,NDvL,ArgL,PArgL):-
    check_var(Arg,VarL,DvL,ArgL,VarL1,DvL1,PArg),
    check_vars(Args,VarL1,NVarL,DvL1,NDvL,PArg,PArgL).

check_var(User_Arg,VarL,DvL,ArgL,[Pvar|VarL],
          DvL1,[User_Arg|ArgL]):-
    name(User_Arg,[100,118|_]),
    change_var(User_Arg,Pvar,DvL,DvL1).
check_var(User_Arg,VarL,DvL,ArgL,VarL,DvL,[User_Arg|ArgL]).
```

4.3.3 การเปลี่ยนเทอมเงื่อนไขของโคอาร์ดิเนตที่กำหนดไปเป็นเทอมของระบบ

หัวข้อนี้เป็นหัวข้อที่สำคัญและมีการใช้ในหลายๆ โมดูลของโปรแกรมดังนั้นผู้เขียนจึงขอแยกอธิบายไว้ต่างหากเพื่อความสะดวกในการอ้างอิงภายหลัง

เงื่อนไขที่ผู้ใช้ระบุเพิ่มเติมในการสอบถามนั้นไม่สามารถจะนำมาใช้ในการสร้างคำถามได้ทันทีในระบบการทำงานของโพรลอค เพราะต้องมีการแปลงตัวแปรโคอาร์ดิเนตที่มีความสัมพันธ์กับเทอมต่างๆ ของ diary/3 และเทอมอื่นๆ ให้สอดคล้องกันก่อนที่จะนำไปรวมกับเงื่อนไขของคำถามภาษาโพรลอค การเปลี่ยนเทอมโคอาร์ดิเนตเรียกใช้เพรตคิเคต dv_term/4(+,-,+,-)

ตัวอย่าง การเรียกใช้งานเพรตคิเคต dv_term/4 โดยใช้ข้อมูลจากภาพประกอบ 4.5.ก และค่าของ DvL ซึ่งส่งเข้าไปได้มาจากการทำงานของ check_var0/5 ข้างบน

```
?- dv_term((contains("MML",dvTopic);
           contains("MML",dvComment)),Term,DvL,NDvL).
Term=(contains([77,77,76],_x21380);
       contains([77,77,76],_x21341)))
NDvL=[dv(_x21236,dvWho),dv(_x21342,dv1),
       dv(_x21343,dv2),dv(_x21344,dv3),
       dv(_x21345,dv4),dv(_x21346,dv5),
       dv(_x21419,dvWhere),dv(_x21380,dvTopic),
       dv(_x21341,dvComment),dv(_x21347,dv6)].
```

โดยเทอมที่ 1 คือ เทอมเงื่อนไขของผู้ใช้ เทอมที่ 2 (Term) คือ เทอมของระบบที่ได้และสอดคล้องเทอมที่ 1 เทอมที่ 3 (DvL) คือ ลิสท์ของตัวแปรและชื่อของตัวแปร ถ้าชื่อตัวแปรใดเคยมีการกำหนดตัวแปรให้เรียบร้อยแล้วก็จะเรียกตัวแปรนั้นมาใช้งานได้เลย ไม่ต้องมีการกำหนดขึ้นใหม่ แต่ถ้ามีการกำหนดขึ้นใหม่ก็จะเพิ่มเข้าไปในลิสท์เดิม เทอมที่ 4 (NDvL) ลิสท์ของตัวแปรทั้งหมด

กำหนดให้ BNF ของเทอมในโพรลอคสำหรับเงื่อนไขของโคอาร์ดิเนตเป็นดังนี้

```
term ::= functor(' args ') | dvar | var | atom | number | list
args ::= term (',' term)
dvar ::= 'dv' name
name ::= string number
list ::= '[' ']' | '[' args '[' (dvar|var) ']' ']'
```

โค้ดการทำงานของ dv_term/4

```

dv_term(Dvar, Var, Vs, Vs1):-
    atom(Dvar), % Is Dvar atom
    dvar(Dvar), !, % Is Dvar diary variable
    change_var(Dvar, Var, Vs, Vs1).
    % conver Dvar to system variable
dv_term(Var, Var, Vs, Vs):-
    var(Var), !. % It is system variable
dv_term(Num, Num, Vs, Vs):-
    number(Num), !. % It is number
dv_term(Atom, Atom, Vs, Vs):-
    atom(Atom), !. % It is atom
dv_term(List, NList, Vs, Vs1):-
    islist(List), !, % It is list
    dvList(List, NList, Vs, Vs1).
dv_term(Term, NTerm, Vs, Vs1):-
    Term=..[Functor|Args], % It is term
    dvArgs(Args, NArgs, Vs, Vs1),
    NTerm=..[Functor|NArgs].

dvar(Dvar):-
    name(Dvar, [100,118|_]).

islist([]).
islist([_|_]).

change_var(Dvar, Var, Vs, Vs):-
    member(dv(Var, Dvar), Vs).
change_var(Dvar, Var, Vs, [dv(Var, Dvar)|Vs]).

dvList([], [], vs, vs).
dvList(Dvar, Var, Vs, Vs1):-
    atom(Dvar), dvar(Dvar), !,
    change_var(Dvar, Var, Vs, Vs1).
dvList(Var, Var, Vs, Vs):-
    var(Var), !.
dvList(Num, Num, Vs, Vs):-
    number(Num), !.
dvList(Atom, Atom, Vs, Vs):-
    atom(Atom), !.
dvList([Arg|Rest], [NArg|NRest], Vs, Vs2):-
    dv_term(Arg, NArg, Vs, Vs1),
    dvList(Rest, NRest, Vs1, Vs2).

dvArgs([], [], Vs, Vs).
dvArgs([Arg|Rest], [NArg|NRest], Vs, Vs2):-
    dv_term(Arg, NArg, Vs, Vs1),
    dvArgs(Rest, NRest, Vs1, Vs2).

```

4.4 การเปลี่ยนแปลงข้อเท็จจริง diary/3

การเปลี่ยนแปลงข้อเท็จจริง diary/3 ผู้ใช้ทำงานโดยผ่านฟอร์ม change.htm (แสดงดังภาพประกอบ ก.2) มีการทำงาน 3 อย่างคือ เพิ่ม ลบ และ แก้ไข diary/3 ในแฟ้มข้อมูลนามสกุล "db" ของตนเอง มีเพรตติเคดหลักคือ make_operation/3(+,+,+), เพรตติเคด make_record/3(+,+,+) มีหน้าที่ทำการบันทึก, เพรตติเคด make_cancel/2(+,+) มีหน้าที่ทำการลบ และเพรตติเคด make_update/2(+,+) หน้าที่ทำการแก้ไข diary/3 ได้การทำงานของเพรตติเคด make_operation/4

```
make_operation('Add',UserId,Appoint,EDate):-
    make_record(UserId,Appoint,EDate).
make_operation('Delete',UserId,Appoint,_):-
    make_cancel(UserId,Appoint).
make_operation('Update',UserId,Appoint,_):-
    make_update(UserId,Appoint).
```

4.4.1 การเพิ่ม diary/3

The screenshot shows a Netscape browser window titled 'Web Base Logical Diary - Netscape'. The address bar shows the URL 'http://pangha.ct.psu.ac.th/~arrudee/WBLD/diary.html/DiaryFrame.htm'. The main content area displays 'Web-Based Logical Diaries' with a navigation menu. The 'Changing a Diary' section contains the following form fields:

Who	Board of directors	eg Jim, #Jenny Jack (more details)
DatePeriod	10 August 1998 to	
TimePeriod	8.00 to 9.59	
Where	N.125	string
Topic	ordinary meeting	string
Comment	I must report the progressional MHL project	string or link("URL","Text")
Importance	8	
See examples	Add Delete Update Open	

ภาพประกอบ 4.6 แสดงหน้าจอการเพิ่มข้อเท็จจริงที่ 1 ของภาพประกอบ 4.1

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~annudee/WBLD/diary.htm/DiaryFrame.htm

Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing a Diary

Who	#Jenny Jack	eg. Jim, #Jenny Jack (more details)
DatePeriod	6 August 1998 to 7 August 1998	
TimePeriod	14.00 to 15.59	
Where	L. 141 Laboratory	string
Topic	MHL Project	string
Comment	Jenny and Jack report the progressional MHL project	string or link("URL", "Text")
Importance	7	
See examples	<input type="button" value="Add"/> <input type="button" value="Delete"/> <input type="button" value="Update"/> <input type="button" value="Clear"/>	

ภาพประกอบ 4.7 แสดงหน้าจอการเพิ่มข้อเท็จจริงที่ 2 และ 3 ของภาพประกอบ 4.1

การเพิ่ม diary/3 ในการทำงานของผู้ใช้โคออร์ก็คือการทำนัดหมายเพิ่มเติม สมมุติผู้ใช้ขณะนี้ คือ John ภาพประกอบ 4.6 John ทำการนัดหมายกับคณะกรรมการบริหารของบริษัท ภาพประกอบ 4.7 John ทำการนัดหมายโคออร์กับ Jenny และ Jack ซึ่งทั้ง Jenny และ Jack ต่างเป็นสมาชิกของระบบแล้ว

การเพิ่ม diary/3 เป็นการกระทำกับเพิ่มข้อมูลสกุล db ผู้ใช้จะต้องใส่ค่าเทอมทั้ง 10 ตัวของ diary/3 ซึ่งกระทำโดยผ่านฟอร์ม change.htm ในฟิลด์ Comment สามารถเชื่อมโยงไปหาเอกสารอื่นๆ ในเว็บได้ ภายใต้โค้ดของ change.htm จะมีโค้ดของ JavaScript¹ ซึ่งทำหน้าที่ในการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะส่งอินพุตข้อมูลทั้งหมดไปให้แก่ Server JavaScript จะกำหนดค่าให้อัตโนมัติถ้าผู้ใช้ไม่ระบุค่าในฟิลด์ Where Topic และ Comment โดยจะกำหนดค่าให้เป็น "no" และ Importance กำหนดให้มีค่าต่ำสุดคือ 1 การเพิ่ม diary/3 จะกระทำได้ที่ต่อเมื่อเวลาที่

¹ JavaScript ในหัวข้อที่ 2.2.4

ระบุนั้นเป็นช่วงเวลาที่ยาว ไม่มีนัดหมายอื่นๆ และไม่สอดคล้องกับกฎ reserved/1 ที่เจ้าของไดอารี่ได้กำหนดไว้

รูปแบบของการเชื่อมโยงไปยังเอกสารอื่นๆ ในเว็บ สำหรับฟิลด์ comment มีดังนี้

```
link("URL","TEXT")
```

URL คือตำแหน่งที่จะเชื่อมโยงไป และ TEXT คือข้อความที่ปรากฏอธิบายลิงค์ ข้อมูลที่เก็บแสดงได้ตั้งชื่อเท็จจริงที่ 5 แต่เมื่อแสดงผลทางหน้าจอ นัดหมายจะเปลี่ยนข้อความดังกล่าวเป็น tag <a href> ของ html เพื่อลิงค์ไปยังเอกสาร ทำงานโดยเรียกใช้เพรตติเคต chk_link/2

ตัวอย่าง การใช้เพรตติเคต chk_link โดยใช้ข้อมูลฟิลด์คอมเมนต์ของชื่อเท็จจริงที่ 5 จากภาพประกอบ 4.1

```
?-chk_link('link("http://pangha.cs.psu.ac.th/~amrudee/
WBLD/movies/ awf.mov","See ")',Tag).
Tag='<a href=" http://pangha.cs.psu.ac.th/~amrudee/
WBLD/movies/ awf.mov ">See</a>'
```

มีโค้ดการทำงานดังนี้

```
chk_link(Comment,Link):-
    extract_url(Comment,URL,Text),
    %The Comment have format link("URL","TEXT")
    link(URL,Text,Link).
chk_link(Comment,Comment).
%The Comment is not have format link("URL","TEXT")
```

การเพิ่มนัดหมายแบ่งเป็น 2 แบบคือ

- นัดหมายภายในไดอารี่ของตนเอง (แสดงดังภาพประกอบ 4.6)
- นัดหมายระหว่างไดอารี่ตนเองกับไดอารี่อื่นๆในระบบ (แสดงดังภาพประกอบ 4.7)

ภายหลังการนัดหมายจะส่ง email ไปแจ้งให้ผู้ที่ถูกนัดหมายทราบโดยอัตโนมัติ

การนัดหมายทั้ง 2 แบบสามารถแบ่งออกเป็น 2 ประเภทย่อยลงไป ตามจำนวนวันการนัดหมายเป็น 2 พวก ประเภทที่ 1 คือนัดหมายที่เสร็จสิ้นในวันเดียว (ภาพประกอบ 4.6) ประเภทที่ 2 คือนัดหมายที่ใช้ช่วงเวลาติดต่อกันหลายวัน (ภาพประกอบ 4.7) จะมีฟิลด์วันสิ้นสุดการนัดหมายให้ผู้ใช้ระบุ เพรตติเคตหลักในการทำงานคือ make_record/3(+,+,+)

```

% make_record between serveral diaries and many days
make_record01(UserId,diary(Who,Time,Item),EDate):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),
    are_diaries_free([UserId|DiariesL],Time,EDate),!,
    make_records_ndiaries(UserId,DiariesL,
        diary(Who,Time,Item),EDate).
% ERROR: have name of diary owner in the list of diary name
make_record01(UserId,diary(Who,_,_),):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),member(UserId,DiariesL),!,
    write_error(4,UserId,DiariesL).
% ERROR: no diary name in the database
make_record01(UserId,diary(Who,_,_),):-
    extract_diaries_name(Who,DiariesL),
    not have_diaries_name(DiariesL),!,
    write_error(1,UserId,DiariesL).
% ERROR: The diaries have not free time for make appointment
make_record01(UserId,diary(Who,Time,_) ,EDate):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),
    not are_diaries_free([UserId|DiariesL],Time,EDate),!,
    write_error(3,UserId,DiariesL).
%make record on your diary and many days
make_record01(UserId,diary(Who,Time,Item),EDate):-
    not extract_diaries_name(Who,_) ,
    are_diaries_free([UserId],Time,EDate),!,
    make_records_ldiary(UserId,diary(Who,Time,Item),EDate).
%ERROR : If all above rule are failed
make_record01(UserId,_,):-
    write_error(2,UserId,_).
%make record between serveral diaries and 1 day.
make_record02(UserId,diary(Who,Time,Item)):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),
    not member(UserId,DiariesL),
    are_diaries_free([UserId|DiariesL],Time),!,
    make_record_ndiaries(UserId,DiariesL,
        diary(Who,Time,Item)).
% ERROR: have name of diary owner in the list of diary name
make_record02(UserId,diary(Who,_,_)):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),member(UserId,DiariesL),!,
    write_error(4,_,_).
% ERROR: no diary name in the database
make_record02(UserId,diary(Who,_,_)):-
    extract_diaries_name(Who,DiariesL),
    not have_diaries_name(DiariesL),!,
    write_error(1,UserId,DiariesL).
% ERROR: The diaries have not free time for make appointment
make_record02(UserId,diary(Who,Time,_)):-
    extract_diaries_name(Who,DiariesL),
    have_diaries_name(DiariesL),
    not are_diaries_free([UserId|DiariesL],Time),!,
    write_error(3,UserId,DiariesL).
%make record on your diary and 1 days
make_record02(UserId,diary(Who,Time,Item)):-
    not extract_diaries_name(Who,_) ,
    are_diaries_free([UserId],Time),!,
    make_record_ldiary(UserId,diary(Who,Time,Item)).
%ERROR : If all above rule are failed
make_record02(UserId,_) :-
    write_error(2,UserId,_).

```


make_reocord01/3 สำหรับนัดหมายที่มีช่วงหลายวันติดต่อกัน make_reocord02/2 สำหรับนัดหมายเดียวใน 1 วัน

extract_diaries_name/2(+,-) ทำหน้าที่สกัดฟิลด์ Who ให้อยู่ในลักษณะชื่อโคอาร์ ถ้าหากขึ้นต้นด้วยสัญลักษณ์พิเศษ '#'

ตัวอย่าง การใช้งานเพรตดิเคต extract_diaries_name/2

```
?- extract_diaries_name('#Jenny Jack',Names).
Names=['Jenny','Jack']
```

have_diaries_name/1 ใช้ตรวจสอบว่ามีโคอาร์เหล่านั้นอยู่ในระบบทั้งหมดจริงหรือไม่
are_diaries_free/3 และ are_diaries_free/2 ใช้ตรวจสอบว่ามีโคอาร์เหล่านั้นว่างตามวันเวลาที่ระบุ โดยมีความสัมพันธ์กับ diary/3 และ reserved/1

มีโค้ดการทำงานดังนี้

```
are_diaries_free([],_).
are_diaries_free([DName|DRest],time(BTm,ETm,Date)):-
    file_name(DName,DNameF,CNameF),
    % file name of facts and rule
    % load facts and rule into the memory
    load_diary_clause(DNameF,CNameF),!,
    is_free_time(BTm,ETm,Date),
    unload_diary_clause(DNameF,CNameF),!,
    are_diaries_free(DRest,time(BTm,ETm,Date)).

are_diaries_free([],_,_).
are_diaries_free([DName|DRest],time(BTm,ETm,Date),EDate):-
    file_name(DName,DNameF,CNameF),
    load_diary_clause(DNameF,CNameF),!,
    are_free_time(BTm,ETm,Date,EDate),
    unload_diary_clause(DNameF,CNameF),!,
    are_diaries_free(DRest,time(BTm,ETm,Date),EDate).

are_free_time(,_,BDate,EDate):-
    after(BDate,EDate).
are_free_time(BTm,ETm,BDate,EDate):-
    is_free_time(BTm,ETm,BDate),
    next_day(BDate,NDate),
    are_free_time(BTm,ETm,NDate,EDate).

is_free_time(BTm,ETm,Date):-
    not is_busy_time(BTm,ETm,Date).

are_busy_time(BTm,ETm,BDate,EDate):-
    not(are_free_time(BTm,ETm,BDate,EDate)).
```

```

is_busy_time(BT,ET,Date):-
    diary(_,time(BT1,ET1,Date),_),
    overlap_timeslot(tm(BT,ET),tm(BT1,ET1),_).
is_busy_time(BT,ET,Date):-
    reserved(time(BT2,ET2,Date)),
    are_integer([BT,ET,BT2,ET2]),
    overlap_timeslot(tm(BT,ET),tm(BT2,ET2),_).
is_busy_time(BT,ET,Date):-
    findall(Date,reserved(time(BT,ET,Date)),Dl),
    length(Dl,L), L > 0.

```

make_records_ndiaries/3(+,+,+) ทำการนัดหมายระหว่างหลายไดอารี่ สำหรับนัดหมายหลายวัน
make_records_ldiary/3(+,+,+) ทำการนัดหมายเฉพาะไดอารี่ของเจ้าของ สำหรับนัดหมายหลายวัน
make_record_ndiaries/3(+,+,+) ทำการนัดหมายระหว่างหลายไดอารี่ สำหรับนัดหมาย 1 วัน
make_record_ldiary/3 (+,+,+) ทำการนัดหมายเฉพาะไดอารี่ของเจ้าของ สำหรับนัดหมาย 1 วัน

การจัดส่ง email ให้กับเจ้าของไดอารี่คนอื่นที่ถูกนัดหมาย เพื่อแจ้งให้ทราบว่ามีการนัดหมายใหม่เกิดขึ้นกับไดอารี่ของคุณ จะเรียกใช้เพอร์ติเคิล mail_to/2(+,+) ซึ่งข่าวสารที่ส่งไปให้จะบันทึกไว้ในแฟ้มชื่อ ../logical-diary/tmp/message.mail

มีโค้ดการทำงานดังนี้

```

mail_to([],_).
mail_to([User|Users],Subject):-
    user(User,Account),
    make_cmd(['mailx'," -s ",
             Subject,"-r ",'WBLgDries@hotmail.com ',
             Account,"<"," ../tmp/message.mail'],
            SendMail),
    system(SendMail),
    mail_to(Users,Subject).

```

4.4.2 การลบ diary/3

การลบ diary/3 คือการยกเลิกข้อมูลการนัดหมายที่บันทึกไว้ก่อนหน้าแล้ว

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~arnudee/WBLD/diary.html/DiaryFrame.htm

Web-Based Logical Diaries

Invisibility
 Viewing a Diary
 Reserved
 Changing a Diary
 Querying a Diary
 Online Help

Changing a Diary

Who	<input type="text" value="eg Jim, #Jenny Jack (more details)"/>
DatePeriod	<input type="text" value="August"/> 1998 to <input type="text" value=""/>
TimePeriod	<input type="text" value=""/> to <input type="text" value=""/>
Where	<input type="text" value=""/> string
Topic	<input type="text" value=""/> string
Comment	<input type="text" value=""/> string or link("URL", "Text")
Importance	<input type="checkbox"/>
See examples	<input type="button" value="Add"/> <input type="button" value="Delete"/> <input type="button" value="Update"/> <input type="button" value="Cancel"/>

ภาพประกอบ 4.8 แสดงหน้าจอการลบนัดหมายทั้งหมดในเดือนสิงหาคม ค.ศ.1998

ข้อเท็จจริง diary/3 ทุกๆ ตัวที่สอดคล้องกับข้อมูลในชุดที่ใส่จะถูกลบซึ่งอาจจะมากกว่า 1 ข้อเท็จจริงที่สอดคล้อง จากภาพประกอบ 4.8 จะลบข้อเท็จจริงที่เป็นการนัดหมายในเดือน 8 ปี 1998 ซึ่งสอดคล้องกับข้อเท็จจริงทุกข้อเท็จจริง ถ้าข้อมูลนั้นเป็นการนัดหมายระหว่างหลายไดอารี่ โปรแกรมจะตามไปลบในไดอารี่อื่นๆ พร้อมทั้งส่ง email เพื่อแจ้งให้เจ้าของไดอารี่นั้นรับทราบ ข้อเท็จจริงที่ 2,3,4 และ 5 ที่ถูกลบไปเป็นนัดหมายระหว่างไดอารี่ดังนั้นก็จะต้องตามไปลบในไดอารี่ของ Jenny และ Jack ด้วย เทรดคิเคตหลักในการทำงานคือ make_cancel/2(+,+)

มีโค้ดการทำงานดังนี้

```

make_cancel (UserId,Diary):-
write_cancel (UserId,Diary,Ans),
file_name (UserId,DName,_),
cancel (DName,Diary),
output_html ([
    cgi_reply,
    start,
    title ('C H A N G I N G   D I A R Y'),
    '<body><p align="center"><strong><big>
<font color="#0000FF" size="5">
DELETED APPOINTMENT(S) IN ',
    UserId, ''s diary', '\\, '</font></big>
</strong></p>',
    itemize (Ans),
    '</body>',
    end]).

```

write_cancel/3(+,+,-) ทำหน้าที่ในการค้น diary/3 ที่สอดคล้องกับที่ต้องการ (Diary) แล้วเขียนให้อยู่ในรูปภาษา HTML ส่งกลับมาเพื่อแสดงให้ผู้ใช้ทราบว่าการลบ diary/3 ใดไปบ้าง (Ans) cancel/2(+,+) ทำหน้าที่ไปลบ diary/3 ที่สอดคล้องออกจากแฟ้มข้อมูลทั้งหมด มีโค้ดการทำงานดังนี้

```

cancel (DiaryName, DAppoint):-
    load_diary (DiaryName),!,
    findall (DAppoint, DAppoint, DAppoints),
    retractall (DAppoint),!,
    % remove all relevant facts from memory
    findall (diary (W, T, I), diary (W, T, I), Diary1),
    unload_db (DiaryName),
    % write the rest of fact to the file
    tell (DiaryName),
    write_diary (Diary1),
    told,
    delete_between_diary (DAppoints).

```

```

delete_between_diary (DAppoint):-
    between_diary (DAppoint, BtwApps),
    delete_b_diary (BtwApps).

```

```

delete_b_diary ([]).
delete_b_diary ([diary (W, T, I) | BtwApps]):-
    extract_diaries_name (W, DNs),
    deletes (DNs, diary (W, T, I)),
    delete_b_diary (BtwApps).

```

delete_between_diary/1(+) ทำหน้าที่ตรวจสอบนัดหมายระหว่างใครแล้วความไปลบ
นัดหมายนั้นออกจากเพิ่มข้อมูลอื่น พร้อมส่ง email แจ้งให้ทราบ ขั้นตอนการส่ง email คล้ายกับ
การส่ง e-mail เพื่อแจ้งการนัดหมาย

4.4.3 การแก้ไข diary/3

การแก้ไข diary/3 คือการเปลี่ยนแปลงข้อมูลการนัดหมาย

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~arrudee/WBLD/diary.html/DiaryFrame.htm

Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing a Diary

Who	<input type="text"/>	eg Jim, #Jenny Jack (more details)
DatePeriod	15 August 1998 to	<input type="text"/>
TimePeriod	19.00 to 20.59	
Where	Jenny's house	string
Topic	<input type="text"/>	string
Comment	<input type="text"/>	string or link("URL", "Text")
Importance	<input type="text"/>	
See examples	<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>	

ภาพประกอบ 4.9 แสดงหน้าจอการแก้ไขการนัดหมาย

เปลี่ยนข้อเท็จจริงที่ 4 สถานที่จาก V.L ไปเป็นที่บ้านของ Jenny ส่วนข้อมูลอื่นยังคงเดิม

เทอมของ diary/3 ที่ไม่สามารถเปลี่ยนค่าได้คือ Who และ TimePeriod ดังนั้นผู้ใช้ต้อง
ระบุ TimePeriod ให้ครบและถูกต้องส่วนฟิลด์ who ไม่ต้องใส่ค่า และใส่ค่าเฉพาะเทอมที่ต้องการ
จะเปลี่ยนแปลงค่าเท่านั้น เทอมที่ไม่ใส่ค่าแสดงว่าไม่มีการเปลี่ยนแปลงค่า การแก้ไขเปลี่ยนแปลง

จึงทำได้เพียงครั้งละ 1 diary/3 เท่านั้นเพราะ 1 ช่วงเวลาทำได้ 1 นัดหมาย เหตุผลเกิดหลักที่ใช้ในการทำงานคือ make_update/2(+,+)

มีโค้ดการทำงานดังนี้

```

make_update(UserId,Diary):-
    updates(UserId,Diary,Ans),!, /* success */
    output_html([
        cgi_reply,
        start,
        title('C H A N G I N G   D I A R Y'),
        '<body><p><strong><big><font color="#008000">',
        '\\, 'UPDATED APPOINTMENT ',
        '\\, 'Diary : ',UserId,\\,
        '</font></big></strong>',
        itemize(Ans),
        '</body>',
        end]).
make_update(UserId,_):-/* unsuccess */
    output_html([
        cgi_reply,start,
        title('C H A N G I N G   D I A R Y'),
        '<body><p><strong><big><font color="#008000">',
        '\\, 'CAN NOT UPDATE APPOINTMENT IN',
        '\\, 'Diary : ',UserId,\\,
        '</font></big></strong>',
        '</body>',
        end]).

```

updates/3(+,+,-) ทำหน้าที่เปลี่ยนแปลงข้อมูลในแฟ้มข้อมูลแล้วส่งข้อมูลการเปลี่ยนแปลงกลับมาแสดงให้ผู้ใช้ทราบผลการทำงาน (Ans) การทำงานก็คล้ายกับการลบนัดหมาย ถ้าหากเป็นการเปลี่ยนแปลงนัดหมายระหว่างใคร่จะต้องไปแก้ไขในใคร่อื่นๆ ด้วย และส่ง e-mail ไปแจ้งให้ทราบด้วย ทำงานโดยเหตุผล update_between_diary/1(+)

มีโค้ดการทำงานดังนี้

```

updates(UserId,diary(Who,Time,Item),Ans):-
    update_file(UserId,diary(Who,Time,Item),
        NewDiary,OldDiary),
    write_update(NewDiary,OldDiary,Ans),
    update_between_diary(NewDiary).

update_between_diary(diary(Who,Time,Item)):-
    extract_diaries_name(Who,Ws),
    update_other_diaries(Ws,diary(_,Time,Item)).
update_between_diary(diary(Who,_,_)):-
    not extract_diaries_name(Who,_).

```

update_file/4(+,+,+,-)ทำหน้าที่ในการเปลี่ยนแปลงข้อมูลในแฟ้มข้อมูล ซึ่งจะต้องมีการสร้าง diary/3 ใหม่ขึ้นมาที่มีความสัมพันธ์กับค่าของเทอมเดิมที่ไม่มีการเปลี่ยนแปลงและค่าของเทอมใหม่ที่มีการเปลี่ยนแปลงทำงานโดยเรียกใช้เพรดดิคต clone_diary/2(+,-) มีได้การทำงานดังนี้

```

update_file(UserId,diary(Who,Time,Item),
            Diary,diary(W,Time,I)):-
    file_name(UserId,DiaryName,_),
    load_diary(DiaryName),!,diary(W,Time,I),
    clone_diary(diary(Who,Time,Item),
    diary(W,Time,I),Diary),
    retract(diary(W,Time,I)),!,
    findall(diary(X,Y,Z),diary(X,Y,Z),Diary1),
    unload_db(DiaryName),
    tell(DiaryName),
    write_diary(Diary1),nl,
    writeq(Diary),write('.'),nl,nl,
    told.

clone_diary(diary(Who,Time,item(topic
                (Topic,Comment),Where,Imp)),
            diary(W,Time,item(topic(T,C),Wh,Im)),
            diary(W1,Time,item(topic(T1,C1),Wh1,Im1))):-
    c_var(Who,W,W1),c_var(Topic,T,T1),
    c_var(Comment,C,C1),c_var(Where,Wh,Wh1),
    c_var(Imp,Im,Im1).

c_var(V1,V2,V2):-var(V1).
c_var(V1,_,V1).

```

4.5 การกำหนดกฎและขกเลิกกฎ reserved/1 และกฎ invisible/2

กระทำโดยผ่านฟอร์ม reserved.htm (ภาพประกอบ ก.3) และ invisible.htm (ภาพประกอบ ก.4) ผู้เขียนขอรวมการทำงานของ 2 กฎไว้ในหัวข้อเดียวกันเพราะมีหลักการทำงานคล้ายคลึงกัน

4.5.1. การเพิ่มและลบกฎ reserved/1

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~arnudee/WBLD/diary.html/DiaryFrame.htm

Web-Based Logical Diaries

Invisibility
 Viewing a Diary
 Reserved
 Changing a Diary
 Querying a Diary
 Online Help

Changing Reserved

TimePeriod:	<input type="text" value="1"/> <input type="text" value="24"/>	integer 1-24 more details or diary variable	more details
DatePeriod:	<input type="text" value="16"/> // <input type="text" value="8"/> // <input type="text" value="1998"/>	integer 1-31/1-12/1998 or diary variable	more details
Conditions	<input type="text"/>		Don't press enter for new line. See examples!

ภาพประกอบ 4.10.ก แสดงหน้าจอการเพิ่มกฎ reserved/1 กรณีไม่มีเงื่อนไข
กฎ : ห้ามทำการนัดหมายในวันที่ 16 สิงหาคม 1998 ตลอดทั้งวัน

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~amrudee/WBLD/day.html/DiaryFrame.htm

Web-Based Logical Diaries

Invisibility
 Viewing a Diary
 Reserved
 Changing a Diary
 Querying a Diary
 Online Help

Changing Reserved

TimePeriod:	<input type="text" value="dvSTime"/> to <input type="text" value="dvETime"/>	integer 1-24 more details or diary variable	more details
DatePeriod:	<input type="text"/> / <input type="text"/> / <input type="text"/>	integer 1-31/1-12/1998 or diary variable	more details
Conditions	<input type="text" value="(dvSTime>=19, dvETime<=24)"/>		more details

Don't press enter for new line. See examples!

10:00:00 AM

ภาพประกอบ 4.10.ข แสดงหน้าจอการเพิ่มกฎ reserved/1 กรณีมีเงื่อนไข
กฎ : ห้ามทำการนัดหมายในช่วงเวลา 18.00 - 23.59 ของทุกวัน

กฎ reserved/1 (ดูโครงสร้างของกฎจากหัวข้อที่ 4.2.3) คือ กฎที่ผู้ใช้กำหนดขึ้นในการจองเวลาภายในไดอารี่ของตนเพื่อไม่ให้มีการนัดหมายเกิดขึ้น โดยกฎที่สร้างขึ้นนี้จะถูกเก็บลงในแฟ้มข้อมูลของผู้ใช้ที่มีนามสกุล .clause (แสดงดังภาพประกอบ 4.3) ผู้ใช้สามารถทำการกำหนดและยกเลิกได้ เพรดคิเลคที่สำคัญในการทำงานคือ build_reserved/9(+,+,+,+,+,+,+,+) ทำหน้าที่ในการสร้าง reserved/2 จากเทอมที่ผู้ใช้อินพุตเข้ามา และ make_operation/3 พิจารณาว่าจะทำการลบหรือเพิ่ม reserved/1 จากหรือลงแฟ้มข้อมูล

ตัวอย่าง การทำงานของเพรดคิเลค build_reserved โดยใช้ข้อมูลจากภาพประกอบ 4.10.ก และ 4.10.ข

```
?-build_reserved(1,24,'$empty',16,8,1998,
                  '$empty',Reserved).
Reserved=reserved(time(1,24,date(16,8,1998)))

?- build_reserved('dvS','dvE','$empty', '$empty',
                  '$empty', '$empty',
                  '$empty',Reserved).
Reserved=reserved(_x12035,_x12040,_x10255):-
_x12035>=19,_x12040=< 24
```

มีได้คการทำงานดังนี้ การทำงานจะคล้ายกับเพรตดิเคต build_diary ที่ได้ธิบายมาแล้วข้างต้น

```
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               (reserved(_):-BodyT)):-
  form_empty_values([BTime,ETime,Time,Day,
                    Month,Year,Date]),
  not_form_empty_values([Body]),!,
  name(Body,Bo1),atom_chars(Bo2,Bo1),
  dv_term(Bo2,BodyT,[],_).
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               reserved(_):-
  form_empty_values([BTime,ETime,Day,Month,
                    Year,Date,Body]),
  not_form_empty_values([Time]),!.
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               (reserved(TT2):-BoT)):-
  form_empty_values([BTime,ETime,Day,Month,Year,Date]),
  not_form_empty_values([Time,Body]),!,
  dv_term(Time,TT2,[],V1),
  name(Body,Bo1),atom_chars(Bo2,Bo1),
  dv_term(Bo2,BoT,V1,_).
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               reserved(T)):-
  form_empty_values([Time,Day,Month,Year,Date,Body]),
  check_empty([BTime,ETime,Date],[BT,ET,Date1]),!,
  dv_term(time(BT,ET,Date1),T,[],_).
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               (reserved(T):-BoT2)):-
  form_empty_values([Time,Day,Month,Year]),
  check_empty([BTime,ETime,Date],[BT,ET,Date1]),
  not_form_empty_values([Body]),!,
  dv_term(time(BT,ET,Date1),T,[],V1),
  name(Body,Bo1),atom_chars(Bo2,Bo1),
  dv_term(Bo2,BoT2,V1,_).
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               reserved(T)):-
  form_empty_values([Time,Date,Body]),
  check_empty([BTime,ETime,Day,Month,Year],
              [BT,ET,D,M,Y]),!,
  dv_term(time(BT,ET,date(D,M,Y)),T,[],_).
build_reserved(BTime,ETime,Time,Day,Month,Year,Date,Body,
               (reserved(T):-BoT)):-
  form_empty_values([Time,Date]),
```

```

check_empty([BTime, ETime, Day, Month, Year], [BT, ET, D, M, Y]),
not_form_empty_values([Body]),!,
dv_term(time(BT, ET, date(D, M, Y)), T, [], V1),
name(Body, Bo1), atom_chars(Bo2, Bo1),
dv_term(Bo2, BoT, V1, _).

```

```

make_operation('MAKE RESERVED', Reserved, UserId):-
    write_reserved(UserId, Reserved).
make_operation('CANCEL RESERVED', Reserved, UserId):-
    delete_reserved(UserId, Reserved).

```

การเพิ่มกฎ Reserved/1

เพรตดิคตหลักที่สำคัญในการทำงานคือ write_reserved(+,+) ทำหน้าที่การบันทึก reserved/1 ที่ได้จากเพรตดิคต build_reserved/9 ลงเพิ่มข้อมูล มีโค้ดการทำงานดังนี้

```

write_reserved(UserId, Reserved):-
    add_ext(UserId, '.clause', DName),
    open(DName, append, Db),
    set_output(Db),
    nl, write(' '),
    writeq(Reserved), write(' ').', nl, nl,
    close(Db).

```

การบันทึกกฎแต่ละกฎ ลงไปในเพิ่มข้อมูลเพื่อสะดวกในการอ่านขึ้นมาใช้งานจะบันทึกกฎอยู่ภายใต้วงเล็บ (reserved/1)

การลบกฎ Reserved /1

คือการยกเลิกกฎ reserved/1 ที่ตั้งไว้แล้วออกจากเพิ่มข้อมูล โดยมีลักษณะการทำงาน เช่นเดียวกับการลบ diary/3 จะลบทุกกฎที่สอดคล้องกับอินพุตที่รับเข้ามา แต่จะไม่มีการสร้างเงื่อนไขสำหรับการลบกฎ เพรตดิคตหลักที่สำคัญคือ delete_reserved/2 (+,+) มีโค้ดในการทำงานดังนี้

```

delete_reserved(UserId, Reserved):-
    add_ext(UserId, '.clause', CName),
    load_clause(CName),
    retractall(Reserved),!,
    findall(c(invisible(W1, T1), B),
        (clause(invisible(W1, T1), B),
         not invisible(who, time)), Invs),!,
    findall(c(reserved(H1), B1), (clause(reserved(H1), B1),
        not reserved(time)), Revs),!,
    unload_db(CName),
    tell(CName),
    write_clause(Invs), write_clause(Revs),
    told.

```

ตัวอย่างกฎ reserved/1 เพิ่มเติม

- ห้ามทำการนัดหมายในเวลากลางคืน

```
reserved(time(S1,S2,_):-
    S1 >=18,S2 <=24
```

- ห้ามทำการนัดหมายที่มีช่วงเวลาการนัดหมายมากกว่า 2 ชั่วโมง

```
reserved(time(S1,S2,_):-
    S2 - S1 > 2.
```

- กำหนดกฎภายใน 1 วันห้ามทำนัดหมายเกิน 3 นัดหมาย

```
reserved(time(_,_,Date)):-
    (findall(Who,diary(Who,time(_,_,Date),_),Ds),
    length(Ds,Ls),Ls >=3).
```

- ห้าม Jack ทำการนัดหมายกับคุณ

```
reserved(_):-
    (getenvstr('REMOTE_USER',User),name('Jack',User)).
```

- ห้ามทำการนัดหมายในวันหยุดสุดสัปดาห์

```
reserved(time(_,_,dvDate)):-
    (date2day(dvDate,dvDay),
    (dvDay = sun ; dvDay=sat)).
หรือ
```

```
reserved(time(_,_,Date)):-
    (date2day(Date,Day),
    member(Day ,[sun , sat])).
```

- ต้องทำการนัดหมายล่วงหน้า 2 วันก่อนวันนัดหมายจริง

```
reserved(time(_,_,Date)):-
    make_before(2,Date).
```

หมายเหตุ : make_before/2 เป็น builtin predicate

```
make_before(N, Date):-
    today(Today),
    next_nday(Today,N,NDay),
    not after(Date,NDay).
```

4.5.2 การเพิ่มและลบกฎ Invisible/2

Web Base Logical Diary - Netscape

http://panha.cs.psu.ac.th/~annudee/WBLD/day.htm/DayFrame.htm

Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing Invisibility

Who: eg. John, Jenny, Jack and all or diary variable such as drWho

TimePeriod: to integer 1-24 or diary variable [more details](#)

DatePeriod: / / integer 1-31/1-12/1998 or diary variable [more details](#)

Conditions:

Don't press enter for new line.
See examples |

ภาพประกอบ 4.11.ก แสดงการเพิ่มกฎ invisible/2 กรณีไม่มีเงื่อนไข

กฎ : ห้ามทุกคนดูนัดหมายในวันที่ 10 สิงหาคม 1998 เวลา 8.00-9.59 น.

Web Base Logical Diary - Netscape

http://pangha.ct.psu.ac.th/~arrudee/WBLD/day.html/DayFrame.htm

Web-Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing Invisibility

Who	<input type="text" value="Jack"/>	<small>eg. John, Jenny, Jack and all or diary variable such as dvWho</small>
TimePeriod:	<input type="text"/> to <input type="text"/>	<small>integer 1-24 or diary variable</small>
DatePeriod:	<input type="text"/> // <input type="text"/> // <input type="text"/>	<small>integer 1-31/1-12/1998 or diary variable</small>
Conditions	<input style="width: 100%;" type="text" value="(date2day(dvDate, dvDay), (dvDay=sat; dvDay=sun))"/>	

[more details](#)

[more details](#)

Don't press enter for new line.
[See examples I](#)

ภาพประกอบ 4.12.ข แสดงการเพิ่มกฎ invisible/2 กรณีมีเงื่อนไข

กฎ : ห้าม Jack ดูนัดหมายในทุกวันเสาร์อาทิตย์

กฎ invisible/2 คือกฎที่ผู้ใช้กำหนดขึ้นเพื่อกำหนดสิทธิ์ในการดูข้อมูลในไดอารี่ของคุณ แต่เจ้าของไดอารี่สามารถมองเห็นข้อมูลทั้งหมดได้ กฎ Invisible/2 จัดเก็บอยู่ในแฟ้มข้อมูลที่มีนามสกุล .clause เปรตคติเกิดหลักในการทำงานคือ build invisible/10(+,+,+,+,+,+,+,+,+) เพื่อสร้าง invisible/2 จากอินพุตที่รับเข้ามาและ make_operation/2 (+,+) ทำการเพิ่มหรือลบ invisible/2 มีได้การทำงานดังนี้

```

% -----
% build invisible/2 clause
% invisible(Who,TimePeriod).
% -----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
invisible(W2,T2)):-
form_empty_values([BTime,ETime,Day,Month,
Year,Date,Body]),!,
check_empty([Who,Time],[Who1,Time1]),
dv_term(Who1,W2,[],V1),
dv_term(Time1,T2,V1,_).

```

```

%-----
% build invisible/2 clause
% invisible(Who,TimePeriod):-Conditions.
%-----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
                (invisible(W2,T2):-BT)):-
    form_empty_values([BTime,ETime,Day,Month,Year,Date]),!,
    not_form_empty_value(Body),!,
    check_empty([Who,Time],[Who1,Time1]),
    dv_term(Who1,W2,[],V1),
    dv_term(Time1,T2,V1,V2),
    name(Body,Bo1),atom_chars(Bo2,Bo1),
    dv_term(Bo2,BT,V2,_).
%-----
% build invisible/2 clause
% invisible(Who,time(BT,ET,Date)).
%-----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
                invisible(W2,T)):-
    form_empty_values([Time,Day,Month,Year,Body]),!,
    check_empty([Who],[Who1]),
    dv_term(Who1,W2,[],V1),
    check_empty([BTime,ETime,Date],[BT,ET,Date1]),
    dv_term(time(BT,ET,Date1),T,V1,_).
%-----
% build invisible/2 clause
% invisible(Who,time(BT,ET,Date)):-Conditions.
%-----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
                (invisible(W2,T):-BTT)):-
    form_empty_values([Time,Day,Month,Year]),!,
    not_form_empty_value(Body),
    check_empty([Who],[Who1]),
    dv_term(Who1,W2,[],V1),
    check_empty([BTime,ETime,Date],[BT,ET,Date1]),
    dv_term(time(BT,ET,Date1),T,V1,V2),
    name(Body,Bo1),atom_chars(Bo2,Bo1),
    dv_term(Bo2,BTT,V2,_).
%-----
% build invisible/2 clause
% invisible(Who,time(BT,ET,date(Day,Month,Year))).
%-----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
                invisible(W2,T)):-
    form_empty_values([Time,Date,Body]),!,
    check_empty([Who],[Who1]),
    dv_term(Who1,W2,[],V1),
    check_empty([BTime,ETime,Day,Month,Year],
                [BT,ET,D,M,Y]),
    dv_term(time(BT,ET,date(D,M,Y)),T,V1,_).
%-----
% build invisible/2 clause
% invisible(Who,time(BT,ET,date(Day,Month,Year))):-Conditions.
%-----
build_invisible(Who,BTime,ETime,Time,Day,Month,Year,Date,Body,
                (invisible(W2,T):-BTT)):-
    form_empty_values([Time,Date]),!,
    not_form_empty_value(Body),!,
    check_empty([Who],[Who1]),
    dv_term(Who1,W2,[],V1),

```

```

        check_empty([BTime, ETime, Day, Month, Year],
                    [BT, ET, D, M, Y]),
        dv_term(time(BT, ET, date(D, M, Y)), T, V1, V2),
        name(Body, Bo1), atom_chars(Bo2, Bo1),
        dv_term(Bo2, BTT, V2, _).

%-----
% 2 operations for invisible/2 clause
%   -MAKE INVISIBILITY
%   -CANCEL INVISIBILITY
%-----

make_operation('MAKE INVISIBILITY', Invisible, UserId):-
    write_invisible(UserId, Invisible).
make_operation('CANCEL INVISIBILITY', Invisible, UserId):-
    delete_invisible(UserId, Invisible).

```

การเพิ่มกฎ Invisible/2

คือการเพิ่มกฎ invisible/2 ลงในแฟ้มข้อมูล .clause เปรตติเคตที่สำคัญในการทำงานคือ write_invisible/2(+,+) มีการทำงานคล้ายกับ write_reserved/2

มีโค้ดการทำงานดังนี้

```

write_invisible(UserId, Invisible):-
    add_ext(UserId, '.clause', DName),
    open(DName, append, Db),
    set_output(Db),
    nl, write(' '),
    writeq(Invisible), write(').'),
    nl, nl,
    close(Db).

```

การลบกฎ Invisible/2

คือการยกเลิกกฎ invisible/2 ที่ตั้งไว้ออกจากแฟ้มข้อมูล โดยมีลักษณะการทำงานเช่นเดียวกับการลบ diary/3 และการลบ reserved/2 แต่จะไม่มีการสร้างเงื่อนไขสำหรับการลบกฎ เปรตติเคตหลักในการทำงานคือ delete_invisible/2(+,+) มีโค้ดการทำงานดังนี้

มีโค้ดการทำงานดังนี้


```

delete_invisible(UserId, Invisible):-
    add_ext(UserId, '.clause', CName),
    load_clause(CName),
    retractall(Invisible),
    findall(c(invisible(W1, T1), B),
            (clause(invisible(W1, T1), B),
             not invisible(who, time)), Invs),
            findall(c(reserved(H1), B1),
                    (clause(reserved(H1), B1),
                     not reserved(time)), Revs),
            tell(CName),
            write_clause(Invs),
            write_clause(Revs), told.

```

ตัวอย่างกฎ invisible/2 เพิ่มเติม

- ห้าม John ดูตารางนัดหมายของคุณแต่คนอื่นสามารถดูได้

```
invisible('John',_).
```

- ห้ามทุกๆ คนดูตารางนัดหมายในช่วงวันหยุดสุดสัปดาห์

```
invisible(all,time(_,Date)):-
    (date2day(Date, Day),
     (Day = sun ; Day=sat)).
```

- ห้ามทุกคนไม่สามารถเห็นนัดหมายของคุณในเดือนกุมภาพันธ์ตั้งแต่วันที่ 5 ถึง 12

```
invisible(all,time(1,24,date(X,2,1998))):-
    X>=5,X<12.
```

- ajin ไม่สามารถดูนัดหมายของคนอื่นได้นอกจากนัดหมายของคุณกับ ajin เท่านั้น

```
invisible(ajin,TimePeriod):-
    diary(Name,TimePeriod,_),Name\==ajin.
```

4.6 การดูไดอารีรายเดือน

การดูตารางนัดหมายจะใช้ฟอร์มที่แสดงดังภาพประกอบ ก.5 โดยผู้ใช้ระบุเดือน ปี และชื่อไดอารีที่ต้องการจะดู ข้อมูลนัดหมาย diary/3 ที่แสดงจะถูกกรองผ่านกฎ invisible/2 ยกเว้นเจ้าของไดอารีสามารถดูข้อมูลได้ทั้งหมด

ตัวอย่างเช่น Jack ดูไดอารีของ John ซึ่ง John มีข้อมูลของการนัดตั้งภาพประกอบ 4.2 และ กฎตั้งภาพประกอบ 4.3 ดังนั้น Jack จะไม่เห็นนัดหมายในวันที่ 10 สิงหาคม 1998 เวลา 8.00-9.59 และทุกนัดหมายในวันเสาร์อาทิตย์ ก็ไม่สามารถเห็นได้

การแสดงผลการนัดหมายในรูปแบบปฏิทินรายเดือนเพื่อดูไดอารี่ของตนเอง และไดอารี่
 ของผู้อื่นนั้น กรณีการดูไดอารี่ของผู้อื่นข้อมูลที่จะนำมาแสดงจะต้องผ่านการคัดเลือกจากกฎ
 invisible/2 ที่เจ้าของไดอารี่กำหนดเสียก่อน โดยเรียกใช้เพรตคิเคต can_sec/2

เพรตคิเคตหลักที่สำคัญในการทำงานคือ view_month/4(+,+,+,+) แต่ละกฎขึ้นกับผู้ดู
 และไดอารี่ที่ดู

Viewing a Diary

DiaryName: eg. Jack, Jenny, John or space for viewing your diary

Month:

Year:

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~amudee/WBLD/diary/html/DiaryFrame.htm

John's Diary

August 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6 **14.00-15.59: #Jenny Jack: MML Project L.141	7 **14.00-15.59: #Jenny Jack: MML Project: L.141	8
				Laboratory: Jenny and Jack report progressional MML project 7	Laboratory: Jenny and Jack report progressional MML project 7	
9	10	11	12	13	14	15
16	17	18	19	20	21	22

ภาพประกอบ 4.12 แสดงการดูไดอารี่ของ John โดย Jack

มีได้การทำงานดังนี้

```

view_month (UserId, UserId, Month, Year) :-
    write_cal0 (UserId, Month, Year, Calendar, DiaryFact, MthStr),
    write_calendar_html (UserId, MthStr, Year, Calendar, DiaryFact).
view_month (UserId, DName, Month, Year) :-
    form_empty_value (DName), !,
    write_cal0 (UserId, Month, Year, Calendar, DiaryFact, MthStr),
    write_calendar_html (UserId, MthStr, Year, Calendar, DiaryFact).

view_month (_, DName, Month, Year) :-
    user (DName, _), !,
    write_cal0 (DName, Month, Year, Calendar, DiaryFact, MthStr),
    write_calendar_html (DName, MthStr, Year,
        Calendar, DiaryFact).
view_month (_, DName, _, _) :-
    write_error (DName).

```

เพรคคิเคต write_cal0/6(+,+,+,-,-) ใช้สำหรับค้นหาข้อมูลไดอารี่ทั้งหมดภายในเดือนนั้นแล้วเขียนในรูปภาษา HTML เป็นตารางของปฏิทิน (Calendar) และ diary/3 ทั้งหมดที่สามารถแสดงได้ (DiaryFact) และชื่อเต็มของเดือน (MthStr) เพรคคิเคต write_calendar_html นำผลลัพธ์จาก write_cal0/6 ไปแสดงส่งกลับไปให้ผู้ใช้โดยซ่อน DiaryFact ไว้ภายใต้ Comment tag เพื่อโปรแกรม Agent จะสามารถนำไปใช้งานได้¹ มีได้การทำงานดังนี้

```

write_cal0 (UserId, Month, Year, Calendar, DiaryFact, MthStr) :-
    add_ext (UserId, '.db', DName1),
    add_ext (UserId, '.clause', CName1),
    load_clause (CName1),
    load_diary (DName1),
    daysinmonth (Month, Year, EDay),
    day_of_week (Year, Month, 1, Day),
    dow (FDay, Day),
    write_cal (UserId, date (FDay, Month, Year),
        EDay, Calendar, DiaryFact),
    map_month (MthStr, Month).

```

เพรคคิเคต find_facts/4 ค้นหาข้อมูลนัดหมายภายในเดือนนั้นซึ่งจะเรียกใช้เพรคคิเคต can_sec/2 ด้วยถ้าหากว่าไม่ใช่เจ้าของไดอารี่ มีได้การทำงานดังนี้

¹ดูเพิ่มเติมในหัวข้อ 4.1.2.2

```

find_facts (UserId, Month, Year, DiaryFact) :-
    getenvstr ('REMOTE_USER', UserStr), name (User, UserStr), !,
    findall (diary (X, time (T1, T2, date (Day, Month, Year))), Z),
    diary (X, time (T1, T2, date (Day, Month, Year))), Z), Apps1),
    change_atom_chars (Apps1, Apps2),
    write_facts (Apps2, Apps3),
    build_tag2 (Apps3, DiaryFact).
find_facts (UserId, Month, Year, DiaryFact) :-
    getenvstr ('REMOTE_USER', UserStr), name (User, UserStr),
    UserId \== User, !,
    findall (diary (X, time (T1, T2, date (Day, Month, Year))), Z),
    (diary (X, time (T1, T2, date (Day, Month, Year))), Z),
    can_see (User, time (T1, T2, date (Day, Month, Year))))),
    Apps1),
    change_atom_chars (Apps1, Apps2),
    write_facts (Apps2, Apps3),
    build_tag2 (Apps3, DiaryFact).

```

เพรตคิเคต write_cal/5(+,+,+,-) ใช้สำหรับเขียนตารางปฏิทินรายเดือน ในรูปภาษา HTML พร้อมทั้งข้อมูลการนัดหมายในแต่ละวัน มีโค้ดการทำงานดังนี้

```

write_cal (UserId, date (FDay, Month, Year), EDay, Calendar, DiaryFact) :-
    find_facts (UserId, Month, Year, DiaryFact),
    Endrow is 7 - FDay,
    write_firstrow (UserId, date (FDay, Month, Year),
        Endrow, Cols),
    build_row (Cols, FirstR),
    NtDay is Endrow + 1,
    write_restrrows (UserId, date (NtDay, Month, Year),
        EDay, RestR),
    build_tag2 ([FirstR, RestR], Calendar).

```

เพรตคิเคต write_firstrow/4(+,+,+,-) ทำหน้าที่เขียนแถวแรกของปฏิทิน เพรตคิเคต write_restrrow/4(+,+,+,-) เขียนแถวที่เหลือของปฏิทิน เพรตคิเคต chk_apps/3 จะตรวจสอบนัดหมายแต่ละวันในเดือนนั้นเมื่อทำการเขียนแต่ละวันลงในปฏิทิน ถ้าไม่ไข่เจ้าของโคอาร์ี่จะต้องเรียกใช้เพรตคิเคต can_see/2 เพื่อกรองข้อมูลอีกครั้งหนึ่ง มีโค้ดการทำงานดังนี้

```
chk_apps (UserId, Date, Apps2) :-
    getenvstr ('REMOTE_USER', UserStr), name (UserId, UserStr), !,
    findall (app (X, time (T1, T2), Z),
            diary (X, time (T1, T2, Date), Z), Apps1),
    write_apps (Apps1, Apps2).
chk_apps (Owner, Date, Apps2) :-
    getenvstr ('REMOTE_USER', UserStr), name (User, UserStr),
    Owner \== User, !,
    findall (app (X, time (T1, T2), Z),
            (diary (X, time (T1, T2, Date), Z),
             can_see (User, time (T1, T2, Date))),
            Apps1),
    write_apps (Apps1, Apps2).
```

บทที่ 5

สรุป

งานวิทยานิพนธ์นี้ได้ศึกษาและพัฒนาโปรแกรมลอจิกโคอาร์บินเว็บ ซึ่งเป็นโปรแกรมต้นแบบของโปรแกรมโคอาร์ที่มีการทำงานบนระบบเครือข่ายเอนมูม โดยชอนภาษาเชิงตรรกไว้เบื้องหลังการทำงาน และมีการทำงานร่วมกับ CGI

5.1 ผลการวิจัย

จากการศึกษาและพัฒนาโปรแกรมได้ผลการวิจัยดังนี้

1. ได้โปรแกรมต้นแบบลอจิกโคอาร์บินเว็บที่มีการทำงานบนระบบเครือข่ายเอนมูม และชอนภาษาเชิงตรรกไว้เบื้องหลังการทำงานจากผู้ใช้ภายใต้ระบบเครือข่ายเอนมูม ที่สามารถตอบคำถามที่ซับซ้อนและผู้ใช้สามารถกำหนดกฎในการทำงานสำหรับการแสดงข้อมูลและการกำหนดเวลาสงวนได้ ซึ่งสามารถนำไปใช้งานได้จริง และสามารถนำโปรแกรมนี้ไปพัฒนาต่อไปเพื่อให้มีประสิทธิภาพในการทำงานมากยิ่งขึ้น
2. ผู้พัฒนาได้เทคนิคและวิธีการชอนการทำงานของภาษาเชิงตรรกไว้เบื้องหลังการทำงานของระบบเครือข่ายเอนมูม โดยใช้ CGI
3. ผู้พัฒนาได้ศึกษาและใช้ระบบความปลอดภัยของเครือข่ายเอนมูมร่วมกับโปรแกรมที่พัฒนาขึ้น

5.2 อุปสรรคและปัญหาในการวิจัย

ในการศึกษาและพัฒนาโปรแกรมมีปัญหาในการทำวิจัยดังนี้

1. เครื่อง Server pangha ที่ใช้ในการทำงานมีฮาร์ดดิสก์น้อยเกินไป เมื่อเนื้อที่เต็มระบบไม่สามารถจะทำงานได้จึงทำให้ต้องทำการแก้ไขเพื่อให้มีเนื้อที่ว่างเพียงพออีกครั้ง
2. เอกสารประกอบสำหรับการใช้งาน BinProlog ที่มีให้ นั้นไม่ดีเท่าที่ควรทำให้ต้องใช้เวลานานในการศึกษาจากคู่มือภาษาโปรแกรมอื่น ๆ เพื่อนำมาประยุกต์ใช้งานกับ BinProlog
3. เนื่องจาก PiLLow Library ที่ดาวน์โหลดมาใช้งานนั้น ไม่ได้พัฒนาขึ้นมาใช้งานกับ BinProlog โดยตรง ดังนั้นจึงต้องมีการแก้ไขและปรับปรุง PiLLow Library ให้สามารถใช้งานร่วมกับ BinProlog ได้ซึ่งต้องใช้เวลานานพอสมควร

4. ปัญหาการสร้างตัวเอ็กซ์ซีคิวทีฟไฟล์ของ BinProlog¹

5.3 ข้อเสนอแนะ

ผู้วิจัยขอแบ่งข้อเสนอแนะเป็น 2 ข้อดังนี้

1. นำโปรแกรมต้นแบบนี้ไปพัฒนาโปรแกรมโคออร์ดิเนตเว็บให้มีประสิทธิภาพในการทำงานให้สมบูรณ์มากยิ่งขึ้น ความสามารถที่สามารถเพิ่มเติมการทำงานของโปรแกรมเช่น การแสดงนัดหมายในรูปแบบอื่นๆ เช่นรายวัน รายสัปดาห์ รายปี การแสดงเวลาสงวน การทำล๊อคไฟล์ เพื่อป้องกันการเข้าใช้ข้อมูลพร้อมกัน และ การเพิ่มเติมกฎอื่นๆ ให้โคออร์ดิเนต เช่น กฎสำหรับค่าเริ่มต้นของระบบ

2. นำวิธีการและความรู้จากการศึกษาและพัฒนา ไปพัฒนาระบบโปรแกรมทางด้านระบบผู้ชำนาญการ และ ปัญญาประดิษฐ์ที่ทำงานบนระบบเครือข่าย เป็นต้น

¹ดูรายละเอียดเพิ่มเติมในหัวข้อ 3.2

บรรณานุกรม

อิสระ ศรีสินทร, 2539, การเขียนโปรแกรมเชิงตรรก (Logic Programming), ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์, มหาวิทยาลัยสงขลานครินทร์.

Apache Group, 1998, Apache Web Server, http://www.apache.org/ABOUT_APACHE.html.

AMF, 1998, AMF Daily Planner&PIM, <http://www.theshareware.net/entries/000001e5.htm>.

Aufrance Associates, 1995, Calendar Internet, <http://highsierra.com/highsierra/calendar.htm>.

Berners-Lee, T., Fielding, R., and Frystyk, H, 1996, HyperText Transfer Protocol (HTTP/1.0)
Specification (RFC 1945), <http://ds.internic.net/rfc/rfc1945.txt>.

Blue Cannon Software, 1996, Calendar Wise, [http://www.softwarelabs.com/business/
business126.htm](http://www.softwarelabs.com/business/business126.htm).

Bratko, I., 1990, Prolog Programming for Artificial Intelligence, second edition, Addison-
Wesley Publishing co.

BrownBear Software, 1998, iCal 2.0, <http://www.brownbearsw.com/ical.htm>.

Cabeza, D., Hermenegildo, M., and Varma, S. 1997. The PiLLoW/CIAO Library for
Internet/WWW Programming, [http://www.clip.dia.fi.upm.es/miscdocs/pillow/
pillow.html](http://www.clip.dia.fi.upm.es/miscdocs/pillow/pillow.html).

EuroSoft, 1997, Calendar Magic, [http://www.focusmm.co.uk/products/productivity/
calendar.html](http://www.focusmm.co.uk/products/productivity/calendar.html).

JSoft Consulting, 1998, Calendar ±, <http://www.jsoftconsulting.com/calendar.htm>.

Logic Pulse Software, 1997, Calendar Quick, <http://www.logicpulse.com/calquick/>.

Lotus Corp, 1997, Lotus Organizer Web Calendar, [http://www.lotus.com/home.nsf/tabs/
calendar](http://www.lotus.com/home.nsf/tabs/calendar).

Malpas, J., 1987, Prolog : A relational language and its applications, Prentice-Hall International Inc.

Pacific Software, 1995, OpenCalendar, <http://dlwest.com/~wmp/Pacific/OpenCalendar.htm>.

Pocket-Sized Software, 1998, Chrono-Logic, http://Softseek.com/Business_and_Productivity/PIMs_Calendare_and_Schedulers/Review_9426_index.html.

Reynolds, M.C. and Wooldridge, A., 1996, Special Edition using JavaScript, QUE.

Sen, S., 1997, Developing an Automated Distributed Meeting Scheduler, IEEE Expert, July/August, 41-45.

Sheridan Software, 1997, Calendar Widgets, <http://www.shersoft.com/products/calendar/calgen.htm>.

Sterling, L., and Shapiro, E., 1986, The Art of Prolog, 2nd edition, MIT Publishing.

Sukpan, A., Santhitham, D., Jirachiefpattana, A. and Davison, A., 1998, A New Time Scheduling Application of the Web using Logic Programming and Agents, INET'98 Conference in Geneva, Switzerland, CD-ROM Version.

Tarau, P., 1997, BinProlog 5.75, <http://clement.info.umoncton.ca/~tarau>.

WebCal Corporation, 1995, WebCal, <http://www.webcal.com/more.html>.

ภาคผนวก ก

ภาพประกอบฟอร์ม HTML ของโปรแกรม

The screenshot shows a Netscape browser window titled "Web Base Logical Diary - Netscape". The main heading is "Web Based Logical Diaries" with a decorative underline. Below the heading is a navigation menu with links: "Invisibility", "Viewing a Diary", "Reserved", "Changing a Diary", "Querying a Diary", and "Online Help". The "Querying a Diary" link is highlighted. The form itself is titled "Querying a Diary" and contains the following fields:

DiaryName:	<input type="text"/>	(* blank for query your diary) eg. Jack, Jenny
Who	<input type="text"/>	eg. 'James', 'Jim' or diary variable
DatePeriod:	<input type="text"/> / <input type="text"/> / <input type="text"/>	integer 1-31/1-12/1998 or diary variable
TimePeriod:	<input type="text"/> to <input type="text"/>	integer 1-24 (more details) or diary variable
Where	<input type="text"/>	string or diary variable
Topic	<input type="text"/>	string or diary variable
Comment	<input type="text"/>	string or diary variable
Importance	<input type="text"/>	integer 1-10 or diary variable
Conditions	<input type="text"/>	Don't press enter for new line. See examples

At the bottom of the form are two buttons: "EXECUTE" and "CLEAR".

ภาพประกอบ ก.1 ฟอร์มการสอบถามไดอารี่ (query.htm)

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~amrudee/WBLD/diary.html/DiaryForms.htm

Web Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing a Diary

Who	<input type="text"/>	eg. Jim, John, #ajin ad duen (more details)
DatePeriod	<input type="text"/> to <input type="text"/>	
TimePeriod	<input type="text"/> to <input type="text"/>	
Where	<input type="text"/>	strings
Topic	<input type="text"/>	strings
Comment	<input type="text"/>	strings or link("URL","Text")
Importance	<input type="text"/>	
See examples	<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Clear"/>	

[Home](#)
[Web Base Log](#)
[Tools](#)
[Help](#)
[Privacy](#)
[Contact Us](#)

ภาพประกอบ ก.2 ฟอรัมการเปลี่ยนแปลง diary/3 (change.htm)

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~arnudee/WBLD/diary.html/DiaryFrame.htm

Web Based Logical Diaries

[Invisibility](#)
[Viewing a Diary](#)
[Reserved](#)
[Changing a Diary](#)
[Querying a Diary](#)
[Online Help](#)

Changing Reserved

TimePeriod:	<input type="text"/>	to	<input type="text"/>	<small>integer 1-24 more details or diary variable</small>	<small>more details</small>		
DatePeriod:	<input type="text"/>	//	<input type="text"/>	//	<input type="text"/>	<small>integer 1-31/1-12/1998 or diary variable</small>	<small>more details</small>
Conditions						<small>Don't press enter for new line. See examples !</small>	
<input type="button" value="MAKE RESERVED"/> <input type="button" value="CANCEL RESERVED"/> <input type="button" value="CLEAR"/>							

ภาพประกอบ ก.3 ฟอรัมการเปลี่ยนแปลงกฎการจองเวลา reserved/1 (reserved.htm)

Web Base Logical Diary - Netscape

http://panpha.cs.psu.ac.th/~arrudoo/WBLD/diary.html/DiaryFrame.htm

Web Based Logical Diaries

[Invisibility](#) @ [Viewing a Diary](#) @ [Reserved](#) @ [Changing a Diary](#) @ [Querying a Diary](#) @ [Online Help](#)

Changing Invisibility	
Who	<input type="text"/> eg. John, jim, all or diary variable such as dvWho
TimePeriod:	<input type="text"/> to <input type="text"/> integer 1-24 or diary variable more details
DatePeriod:	<input type="text"/> // <input type="text"/> // <input type="text"/> integer 1-31/1-12/1998 or diary variable more details
Conditions	<div style="border: 1px solid black; height: 40px; width: 100%;"></div> <div style="text-align: right; padding-right: 10px;"> Don't press enter for new line. See examples ! </div>
<input type="button" value="MAKE INVISIBLE"/> <input type="button" value="CANCEL INVISIBLE"/> <input type="button" value="CLEAR"/>	

ภาพประกอบ ก.4 ฟอรัมการเปลี่ยนแปลงกฎการมองเห็น invisible/2 (invisible.htm)

Web Base Logical Diary - Netscape

http://pangha.cs.psu.ac.th/~anrudoc/WBLD/diary.html/DiaryFrame.htm

Web Based Logical Diaries

[Invisibility](#) [Viewing a Diary](#) [Reserved](#) [Changing a Diary](#) [Querying a Diary](#) [Online Help](#)

Viewing a Diary

DiaryName (* blank for view your diary) eg. Jack, Jenny, Jim

Month

Year

ภาพประกอบ ก.5 ฟอรัมแสดงการนัดหมายรายเดือน (view.htm)

ภาคผนวก ข

การติดตั้งระบบ

ข.1 ความต้องการของระบบ

แบ่งความต้องการเป็น 2 ด้านดังนี้

ข.1.1 ด้านเซิร์ฟเวอร์

1. เครื่องคอมพิวเตอร์ SUN SPARC station หน่วยความจำ 32 MB ฮาร์ดดิสก์ 1.05 GB ระบบปฏิบัติการยูนิกซ์ Solaris 2.X เพื่อเป็นเซิร์ฟเวอร์ที่ต่อกับระบบเครือข่ายและมีเซิร์ฟเวอร์ HTTP ทำงานอยู่ โปรแกรมเซิร์ฟเวอร์ที่ใช้งานคือ Apache สามารถดาวน์โหลดมาใช้งานได้จาก http://www.apache.org/ABOUT_APACHE.html.

2. ตัวแปลภาษาไพธอน BinProlog สามารถดาวน์โหลดมาใช้งานได้จาก <http://clement.info.umoncton.ca/~tarau>

ข.1.2 ด้านไคลเอนต์

1. เครื่องคอมพิวเตอร์ PC (Personal Computer) ที่สามารถเชื่อมเข้ากับระบบเครือข่าย

2. โปรแกรมบราวเซอร์ที่สามารถทำงานร่วมกับจาวาสคริปต์ได้เช่น เน็ตเคปคอมมูนิเคเตอร์ (Netscape Communicator) อินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer 4.0)

ข.2 ขั้นตอนการติดตั้งโปรแกรมทางด้านเซิร์ฟเวอร์

1. คัดลอกแฟ้มข้อมูลโปรแกรมทั้งหมดลงในเซิร์ฟเวอร์ โดยโครงสร้างของไคลเอนท์ที่แสดงคิงภาพประกอบ ข.2 และ ภาพประกอบ ข.3

สร้างไคลเอนท์ WBLD ภายใต้ไคลเอนท์ที่สามารถให้บริการเอกสาร HTML ได้โดยดูจากแฟ้มข้อมูล `sm.conf` ของเว็บเซิร์ฟเวอร์ว่ากำหนดภายใต้ไคลเอนท์ใด จากภาพประกอบ ข.1 แสดงว่าจะต้องสร้างและคัดลอกไฟล์ทั้งหมดลงในไคลเอนท์ `htdocs` (หรือภายใต้ไคลเอนท์ `public_html` ของผู้ใช้)

☐ สร้างไคเรกทอรี logical-diary ภายใต้ไคเรกทอรีที่สามารถให้บริการเอกสาร CGI Script ได้ จากภาพประกอบ ข.1 แสดงว่าจะต้องสร้างและคัดลอกไฟล์สคริปต์และข้อมูลทั้งหมด ลงภายใต้ไคเรกทอรี cgi-bin

```
# DocumentRoot: The directory out of which you
# will serve your documents.
# By default, all requests are taken from this directory,
# but symbolic links and aliases may be used to point
# to other locations.

DocumentRoot /opt/htdocs

# UserDir: The name of the directory which
# is appended onto a user's home
# directory if a ~user request is recieved.

UserDir public_html

# ScriptAlias: This controls which
# directories contain server scripts.
# Format: ScriptAlias fakename realname

ScriptAlias /cgi-bin/ /opt/httpd/cgi-bin/
```

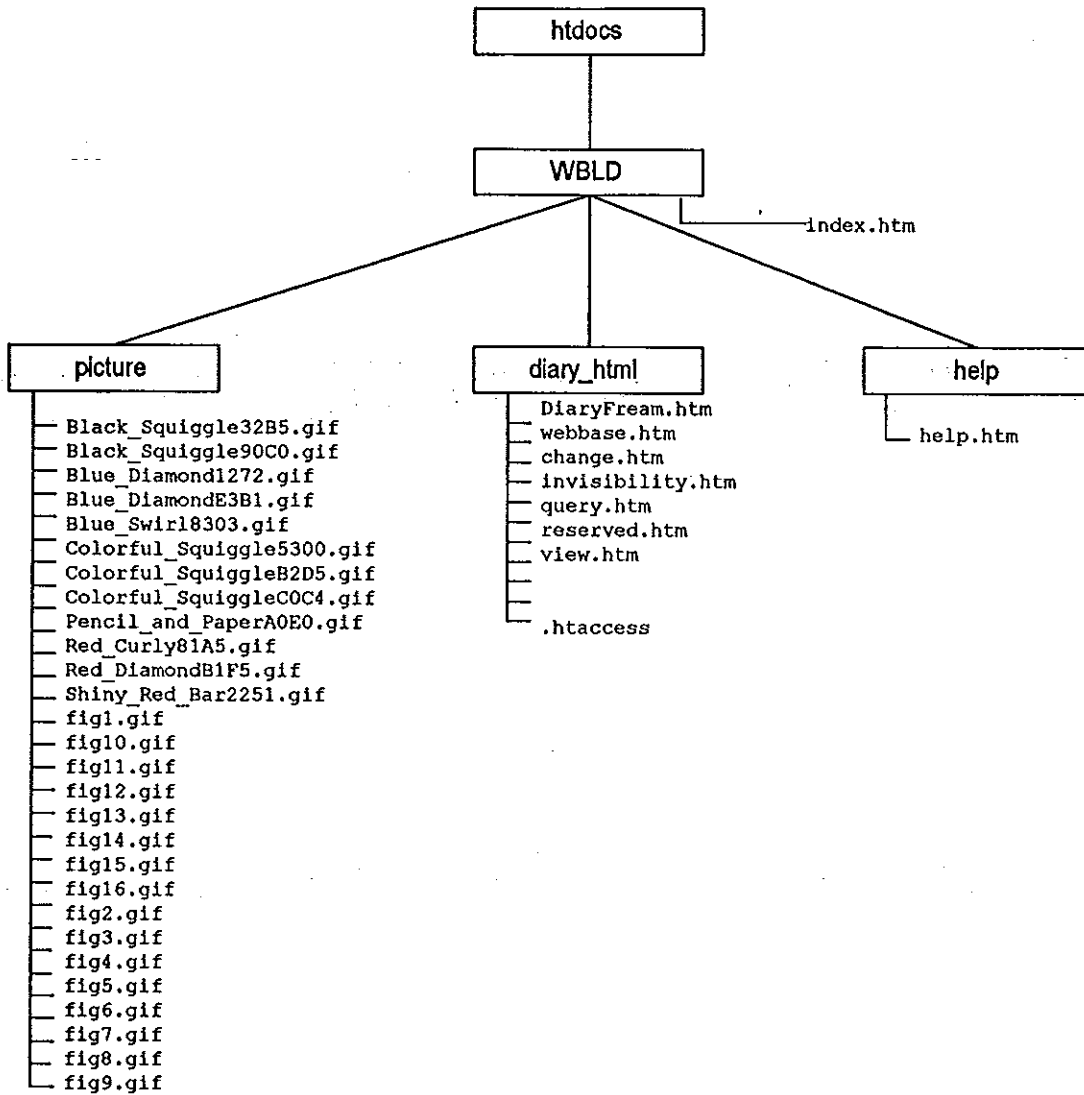
ภาพประกอบ ข.1 ภาพแสดงแก้ไขข้อมูล srm.conf บางส่วน

2. แก้ไขเพิ่มข้อมูล lib.pl ที่เพรคคิเดต url_wbld/1(+)โดยเปลี่ยนเทอมของเพรคคิเดต เป็น URL ที่จะอ้างอิง HTML ของโปรแกรมภายใต้เซิร์ฟเวอร์ที่ใช้งาน

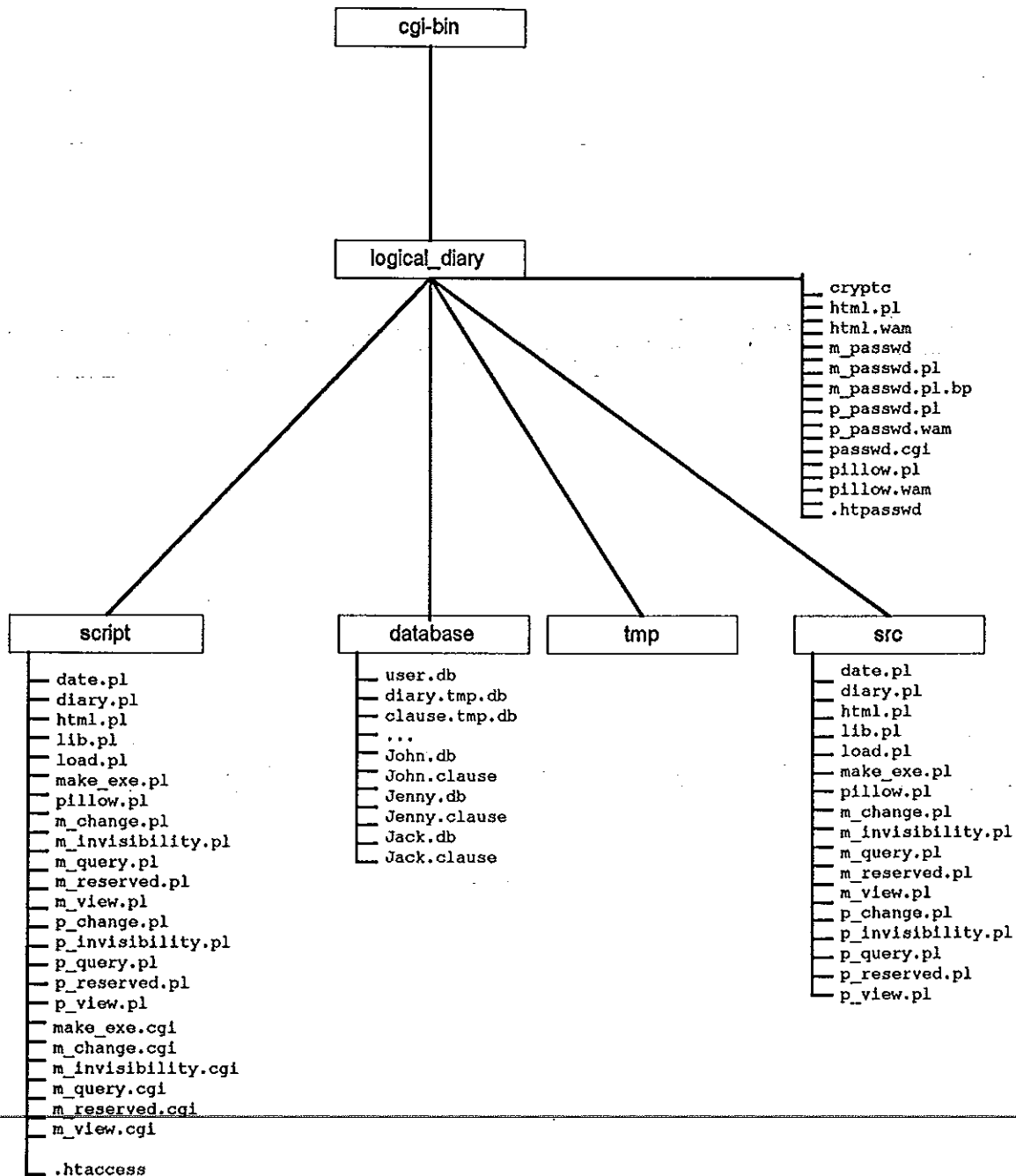
3. สร้างคัมเอ๊กซ์ซีคิวไฟล์ของ CGI Script ใหม่โดยการรันไฟล์ make_exe ภายใต้ไคเรกทอรี logical-diary และไคเรกทอรี script

ตัวอย่าง การทำงานรันไฟล์ make_exe ภายใต้ไคเรกทอรี logical-diary และมีไฟล์ bp อยู่ในไคเรกทอรี /opt/bin

```
% make_exe
Where is bp file : '/opt/bin'. %ผู้ใช้ระบุเส้นทางที่ไฟล์ bp
m_passwd.pl = time(1540) %อยู่ในที่นี้คือ/opt/bin
Succesed make execute cgi files
```

ภาพประกอบ ข.2 โครงสร้างไครกทอรี่ WBLD ภายใต้ไครกทอรี่ htdocs



ภาพประกอบ ข.3 โครงสร้างไดเรกทอรี logical-diary ภายใต้ไดเรกทอรี cgi-bin

ประวัติผู้เขียน

ชื่อ นางสาวเอมฤดี สุขปັນ
วัน เดือน ปีเกิด วันที่ 23 ธันวาคม พ.ศ. 2517

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิทยาศาสตรบัณฑิต	มหาวิทยาลัยสงขลานครินทร์	2538
เกียรตินิยมอันดับ 2 (วิทยาการคอมพิวเตอร์)	วิทยาเขตหาดใหญ่	

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนผลการเรียนดีเด่นของมหาวิทยาลัยสงขลานครินทร์
ทุนอุดหนุนโครงการการพัฒนาซอฟต์แวร์ขนาดเล็กประจำปี 2541 ของศูนย์เทคโนโลยี
อิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ
กระทรวงวิทยาศาสตร์เทคโนโลยีและสิ่งแวดล้อม