หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและแนวความคิดที่ใช้ในงานวิจัย

ในบทนี้จะกล่าวถึงทฤษฎีและแนวคิดต่าง ๆ ที่ใช้ในการวิจัย ได้แก่ ลักษณะของโครงสร้างข้อมูลแบบทู-ทรัย ดีแอลแอล แอดทีฟเอกซ์คอนโทรล รวมทั้งลักษณะและสถาปัตยกรรมของส่วนต่อประสานโอดีบีซี เพื่อใช้เป็นความรู้ในการออกแบบและพัฒนาเครื่องมือให้ทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

2.1 โครงสร้างข้อมูลแบบทู-ทรัย

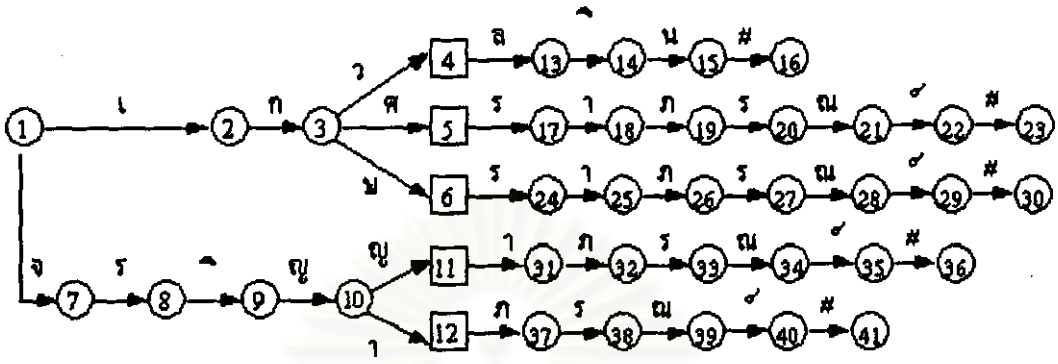
โครงสร้างข้อมูลแบบทู-ทรัย (Two-Trie Structure) ได้พัฒนาขึ้นจากโครงสร้างข้อมูลแบบทรัย (Trie Structure) ลักษณะของโครงสร้างข้อมูลแบบทรัยเหมาะกับข้อมูลที่ประกอบด้วยตัวอักษรเรียงต่อกันไปจึงทำให้สามารถใช้ได้ทั้งข้อความภาษาไทยและภาษาอังกฤษ และเวลาที่ใช้ในการค้นหาข้อมูลน้อยจึงทำให้การค้นหาข้อมูลมีประสิทธิภาพดี[1] จึงทำให้โครงสร้างข้อมูลแบบทรัยได้นำไปใช้ในงานด้านต่าง ๆ เช่น การสร้างพจนานุกรมสำหรับการประมวลผลภาษาธรรมชาติ และการค้นหาคำสงวน (Reserved Words) ของตัวแปลภาษา (Compiler) เป็นต้น โครงสร้างข้อมูลแบบทรัยยังเหมาะกับการค้นหาแบบส่วนเติมหน้า (Prefix Searching) อีกด้วยเนื่องจากโครงสร้างข้อมูลแบบทรัยจะเก็บส่วนเติมหน้า (Prefix) ของแต่ละดัชนีที่เหมือนกันไว้เพียงที่เดียว

โครงสร้างข้อมูลแบบทรัยประกอบด้วยสตริง (String) 2 ส่วน คือ สตริงส่วนหน้า (Front String) และสตริงส่วนหลัง (Rear String) โดยที่สเตท (State) หรือโหนด (Node) และทรานสิชั่น (Transition) ที่มีลักษณะเป็นตัวอักษรของสตริงส่วนหน้าจะใช้ร่วมกันระหว่างดัชนีต่าง ๆ แต่โหนดและทรานสิชั่นของสตริงส่วนหลังจะไม่ใช้ร่วมกันโดยมีเซพพาเรทโหนด (Separate Node) เป็นโหนดที่ใช้แยกความแตกต่างของแต่ละดัชนีที่เก็บในโครงสร้างข้อมูลแบบทรัย ซึ่งคั่นระหว่างสตริงส่วนหน้าและสตริงส่วนหลัง และในสตริงส่วนหลังจะใช้สัญลักษณ์ "#" เป็นโหนดสิ้นสุด (Terminal Node) ของแต่ละดัชนี

---

[1] Thomas A. Standish, Data Structures, Algorithms & Software Principles In C. Addison Wesley Publishing Company, Inc, 1994.

ตัวอย่างข้อมูลที่นำมาสร้างโครงสร้างข้อมูลแบบทรัย ได้แก่ "เกวลิน" "เกศราภรณ์" "เกษราภรณ์ " "จริญญาภรณ์ " และ "จริญาภรณ์" ซึ่งจะได้โครงสร้างข้อมูลแบบทรัยดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างข้อมูลแบบทรัย

จากรูปที่ 2.1 สัญลักษณ์ที่ใช้ในโครงสร้างข้อมูลแบบทรัยได้แก่ โหนดคือวงกลมที่มีตัวเลขกำกับ ส่วนทรานสิชั่นคือตัวอักษรที่อยู่บนเส้นทางระหว่างโหนด 2 โหนดใด ๆ และเซพพาเรทโหนดแทนด้วยสี่เหลี่ยมที่มีตัวเลขกำกับ

การค้นหาดัชนีในโครงสร้างข้อมูลแบบทรัยจะนำอักษรแต่ละตัวในดัชนีมาเปรียบเทียบ ตัวอย่างเช่นต้องการค้นหาดัชนี "เกศราภรณ์" จะนำอักษรตัวแรกมาค้นหาโดยเริ่มจากโหนด 1 ซึ่งอักษรแรก "เ" จะได้เส้นทางจากโหนด 1 ไปโหนด 2 จากนั้นนำอักษรถัดไป "ก" มาค้นหาต่อจากโหนดปัจจุบัน (โหนด 2) จะได้เส้นทางจากโหนด 2 ไปยังโหนด 3 ส่วนอักษรถัดไป "ศ" จะได้เส้นทางจากโหนด 3 ไปยังโหนด 5 ที่เป็นเซพพาเรทโหนดเชื่อมไปยังสตริงส่วนหลัง "ราภรณ์" ซึ่งตรงกับส่วนที่เหลือของดัชนีที่ใช้ในการค้นหา ดังนั้นเวลาที่ใช้ในการค้นหาจึงขึ้นกับจำนวนตัวอักษรที่ประกอบขึ้นมาเป็นดัชนี ไม่ขึ้นอยู่กับจำนวนคำที่เก็บทำให้ค้นหาข้อมูลได้อย่างรวดเร็วเมื่อเทียบกับปริมาณข้อมูลที่จัดเก็บ แต่ข้อเสียของโครงสร้างข้อมูลแบบทรัยคือเมื่อนำไปใช้กับกลุ่มของดัชนี (Key Set) ขนาดใหญ่จะทำให้โครงสร้างข้อมูลแบบทรัยต้องใช้จำนวนโหนดและจำนวนทรานสิชั่นมากจึงทำให้ใช้เนื้อที่มากในการเก็บ และขั้นตอนวิธีในการเพิ่มและลบดัชนีของโครงสร้างข้อมูลแบบทรัย ทำให้เกิดเนื้อที่ว่างจำนวนมาก[2]

---

[2] Jun-ichi Aoe, An Efficient Digital Search Algorithm by Using a Double-Array Structure, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, 1989, pp. 1066-1077.

ต่อมา Jun-ichi Aoe, Katsushi Morimoto, Masami Shishibori และ Ki-Hong Park ได้ร่วมกันคิดค้นโครงสร้างข้อมูลแบบทู-ทรัย[3] ซึ่งสามารถช่วยลดขนาดของโครงสร้างข้อมูลแบบทรัยได้ ซึ่งโครงสร้างข้อมูลแบบทู-ทรัยมีลักษณะดังนี้

1. ข้อมูลของแต่ละดัชนีสามารถระบุได้อย่างเป็นเอกลักษณ์ (Unique)
2. ลดจำนวนทรานสิชั่นโดยให้มีการใช้ทรานสิชั่นร่วมกันในส่วนของสตริงส่วนหน้าและในส่วนของสตริงส่วนหลัง
3. ใช้ได้กับกลุ่มของดัชนีที่มีลักษณะเป็นแบบพลวัต (Dynamic) โดยการใช้ขั้นตอนวิธีในการปรับ (Update) ที่มีประสิทธิภาพในการจัดหน่วยความจำที่ไม่ได้ใช้งานในกรณีที่มีการปรับโครงสร้างข้อมูลแบบทรัย

หลักการของโครงสร้างข้อมูลแบบทู-ทรัยคือ ให้มีการใช้ทรานสิชั่นในสตริงส่วนหลังร่วมกันได้ โดยที่ยังคงความสามารถในการระบุข้อมูลของแต่ละดัชนีได้เพียงหนึ่งเดียว และเวลาที่ใช้ในการค้นหาจะขึ้นกับจำนวนตัวอักษรที่ประกอบขึ้นมาเป็นดัชนี ไม่ขึ้นอยู่กับจำนวนคำที่เก็บ ทำให้ค้นหาข้อมูลได้อย่างรวดเร็วเช่นเดียวกับโครงสร้างข้อมูลแบบทรัย

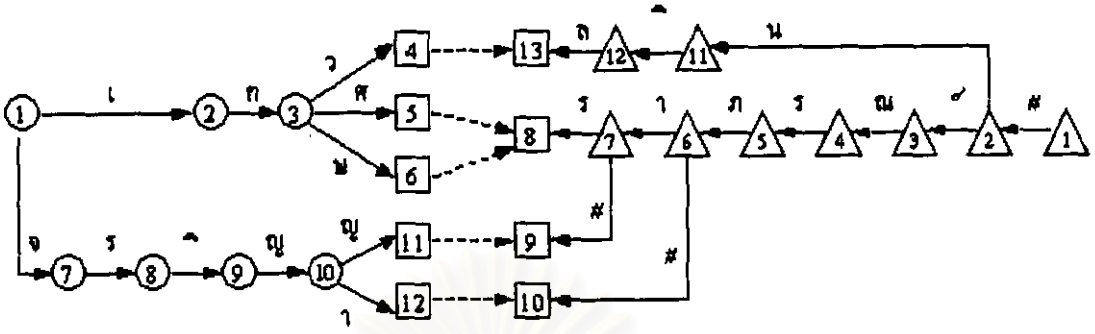
โครงสร้างข้อมูลแบบทู-ทรัยประกอบด้วยโครงสร้างข้อมูลแบบทรัย 2 ส่วนได้แก่ ทรัยสำหรับสตริงส่วนหน้าเรียกว่า ทรัยส่วนหน้า (FR Trie) และทรัยสำหรับสตริงส่วนหลังเรียกว่าทรัยส่วนหลัง (RE Trie) ซึ่งทรัยส่วนหลังจะเก็บสตริงในรูปแบบผันกลับ (Reverse) โดยที่ระหว่างทรัยส่วนหน้าและทรัยส่วนหลังจะมีลิงค์ (Link) ในการเชื่อมจากเซพพาเรทโหนดของทรัยส่วนหน้าไปยังแอคเซพติงโหนด (Accepting Node) ของทรัยส่วนหลัง โดยที่เซพพาเรทโหนดหมายถึงโหนดแรกที่สามารถแยกความแตกต่างของแต่ละดัชนี ดังนั้นข้อมูลของแต่ละดัชนีจึงสามารถผูกติด (Attach) ไว้ที่เซพพาเรทโหนดของทรัยส่วนหน้าได้ ลักษณะของโครงสร้างข้อมูลแบบทู-ทรัยแสดงดังรูปที่ 2.2

จากรูปที่ 2.2 สัญลักษณ์ที่ใช้ในโครงสร้างข้อมูลแบบทู-ทรัยได้แก่ ลิงค์ใช้แทนด้วยเส้นประ โหนดที่อยู่ในกรอบสี่เหลี่ยมแทนแอคเซพติงโหนด (แอคเซพติงโหนดของทรัยส่วนหน้าเรียกว่า เซพพาเรทโหนด) และโหนดที่อยู่ในรูปสามเหลี่ยมแทนโหนดของทรัยส่วนหลัง

---

[3] Jun-ichi Aoe, Katsushi Morimoto, Masami Shishibori, and Ki-Hong Park, A Trie Compaction Algorithm for a Large Set of Keys, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, June 1996, pp. 476-491.

และจะเห็นว่าสตริงที่อยู่ในทรัยส่วนหลังจะเก็บแบบผันกลับเพื่อที่จะลดทรานสิชั่นที่ซ้ำซ้อนกันในกรณีของการเพิ่มและลบดัชนี โดยมี '#' เป็นโหนดสิ้นสุด



รูปที่ 2.2 โครงสร้างข้อมูลแบบทู-ทรัย

ตัวอย่างข้อมูลที่ใช้ในการสร้างโครงสร้างข้อมูลแบบทู-ทรัยในรูปที่ 2.2 จะใช้ข้อมูลเดียวกับที่ใช้สร้างโครงสร้างข้อมูลแบบทรัยในรูปที่ 2.1 ซึ่งจะเห็นว่าจำนวนโหนดและทรานสิชั่นที่ใช้ในโครงสร้างข้อมูลแบบทู-ทรัยน้อยกว่าจำนวนโหนดและทรานสิชั่นในโครงสร้างข้อมูลแบบทรัย

จากการทดสอบในงานวิจัยของ Jun-ichi Aoe, Katsushi Morimoto, Masami Shishibori และ Ki-Hong Park[4] ในการวัดประสิทธิภาพของโครงสร้างข้อมูลทรัยแบบต่าง ๆ โดยใช้กลุ่มของดัชนีที่มีขนาดแตกต่างกัน พบว่าประสิทธิภาพของเวลาที่ใช้ในการค้นหา เพิ่ม และลบดัชนีของโครงสร้างข้อมูลแบบทรัยไม่แตกต่างจากโครงสร้างข้อมูลแบบทู-ทรัยมากนัก แต่ในกรณีที่กลุ่มของดัชนีมีขนาดใหญ่มาก โครงสร้างข้อมูลแบบทู-ทรัยจะใช้เนื้อที่น้อยกว่าโครงสร้างข้อมูลแบบทรัย ดังนั้นโครงสร้างข้อมูลแบบทู-ทรัยจึงเป็นโครงสร้างข้อมูลที่มีประสิทธิภาพ และเหมาะสมกับงานทางด้านค้นหาข้อมูลมากกว่าโครงสร้างข้อมูลแบบทรัย

โครงสร้างข้อมูลทู-ทรัยในเชิงโปรแกรมแบ่งเป็น 2 แบบได้แก่ โครงสร้างเชิงรายการ (List Structure) และโครงสร้างแถวลำดับคู่ (Double-Array Structure)

2.2 ดีแอลแอล

ดีแอลแอล (DLL: Dynamic Link Library) เป็นฟังก์ชันบริการต่าง ๆ บนระบบปฏิบัติการวินโดวส์ ที่ถูกเก็บรวมไว้เป็นไลบรารี (Library) มีลักษณะเป็นมอดูล (Module) ที่ประกอบ

---

[4] Ibid., pp. 487-489.

ด้วยคำสั่งต่าง ๆ แต่ละดีแอลแอลจะมีวัตถุประสงค์ในการทำงานอย่างหนึ่ง โปรแกรมต่าง ๆ สามารถเรียกใช้ดีแอลแอลร่วมกันได้ในขณะที่ดำเนินงาน (Run-Time) เนื่องจากไลบรารีที่เป็นดีแอลแอลนี้มีลักษณะเป็นไฟล์กระทำการ (Executable File) ที่อยู่ในหน่วยความจำ (Memory) เพียงที่เดียวและจะไม่ถูกนำไปรวมกับโปรแกรมที่เรียกใช้เหมือนกับไลบรารีที่เป็นลิงค์แบบสถิต (Static Linking) แต่จะให้โปรแกรมลิงค์ (Link) ไปยังไลบรารีที่ต้องการเมื่อโปรแกรมนั้นถูกเรียกใช้งานดังนั้นไลบรารีเหล่านี้จึงเรียกว่าเป็นลิงค์แบบพลวัต (Dynamic Linking) โปรแกรมประยุกต์หรือดีแอลแอลใดที่มีการทำงานบางส่วนตรงกับดีแอลแอลที่มีอยู่ โปรแกรมหรือดีแอลแอลนั้นสามารถเรียกใช้ดีแอลแอลที่ต้องการได้ และเนื่องจากแต่ละดีแอลแอลมีเพียงสำเนา (Copy) เดียวที่อยู่ในหน่วยความจำจึงทำให้สามารถปรับปรุงหรือแก้ไขดีแอลแอลได้ โดยที่ไม่ต้องทำการเปลี่ยนแปลงโปรแกรมประยุกต์ที่เรียกใช้ดีแอลแอลนั้นตราบใดที่ส่วนต่อประสาน (Interface) ของฟังก์ชันในดีแอลแอลไม่เปลี่ยนแปลง

ระบบปฏิบัติการวินโดวส์จะเก็บดีแอลแอลเหล่านี้ไว้ เมื่อโปรแกรมประยุกต์เริ่มการทำงาน ระบบปฏิบัติการจะนำดีแอลแอลที่โปรแกรมเรียกใช้มาไว้ในหน่วยความจำ เพื่อให้โปรแกรมประยุกต์เรียกใช้ฟังก์ชันต่าง ๆ ในดีแอลแอล โดยที่ฟังก์ชันเหล่านั้นจะใช้สแตค (Stack) (สแตคหมายถึงเนื้อที่การทำงานของโปรแกรมประยุกต์ที่อยู่ในหน่วยความจำ) ของโปรแกรมประยุกต์ในการทำงาน[5] ลักษณะนี้เรียกว่าดีแอลแอลเป็นผู้ให้บริการในโปรเซส (In-Process Server) ซึ่งจะแตกต่างจากผู้ให้บริการนอกโปรเซส (Out-Process Server) ที่ใช้เนื้อที่การทำงานแยกจากโปรแกรมที่เรียกใช้ การทำงานของผู้ให้บริการนอกโปรเซสจะช้ากว่าผู้ให้บริการในโปรเซส เมื่อไม่มีโปรแกรมประยุกต์ใดเรียกใช้ดีแอลแอล ระบบปฏิบัติการจะปลดปล่อย (Unload) ดีแอลแอลนั้นออกไป ทำให้ประหยัดเนื้อที่ในหน่วยความจำ

ผู้พัฒนาซอฟต์แวร์สามารถเพิ่มฟังก์ชันบริการบนระบบปฏิบัติการวินโดวส์ได้โดยการสร้างดีแอลแอลที่มีฟังก์ชันในการทำหน้าที่ต่าง ๆ เพิ่มเติมจากดีแอลแอลที่มีอยู่ ซึ่งจะทำให้โปรแกรมประยุกต์อื่นบนระบบปฏิบัติการวินโดวส์สามารถเรียกใช้ได้ ดีแอลแอลส่วนใหญ่จะใช้นามสกุล (Extension) .DLL แต่อาจจะใช้นามสกุลอื่นได้ เช่น .DRV .FON หรือ .OCX

---

[5] Peter J. Morris, Windows Advanced Programming & Design, Newtech An imprint of Butterworth-Heinemann Ltd, 1993.

ดีแอลแอลนับเป็นคุณสมบัติที่เด่นมากของระบบปฏิบัติการวินโดวส์ เนื่องจากฟังก์ชันจะไม่รวมอยู่กับโปรแกรมประยุกต์ แฟ้มข้อมูลของโปรแกรมประยุกต์จึงมีขนาดเล็ก ใช้เนื้อที่เก็บน้อยกว่าธรรมดา[6]

โครงสร้างข้อมูลแบบทู-ทรีที่นำมาใช้ในการจัดเก็บและค้นหาข้อมูลของเครื่องมือที่พัฒนาขึ้นนี้จะมีลักษณะเป็นดีแอลแอลสำหรับใช้บนระบบปฏิบัติการวินโดวส์ ซึ่งเครื่องมือแบบแอคทีฟเอกซ์คอนโทรลและโปรแกรมประยุกต์อื่นสามารถเรียกใช้ดีแอลแอลนี้ได้

2.3 แอคทีฟเอกซ์คอนโทรล

แอคทีฟเอกซ์คอนโทรล (ActiveX Control) เป็นรูปแบบของออบเจ็กต์ที่เป็นส่วนประกอบ (COM: Component Object Model) ที่ใช้ในการสร้างวินโดวส์คอนโทรล (Windows Control) แบบหนึ่ง

วินโดวส์คอนโทรล คือคอนโทรลต่าง ๆ ที่ได้ออกแบบมาเพื่อใช้สร้างส่วนต่อประสานกับผู้ใช้ในโปรแกรมประยุกต์บนระบบปฏิบัติการวินโดวส์ วินโดวส์คอนโทรลแบ่งเป็น 2 แบบคือ (1)คอนโทรลมาตรฐาน (Standard Control) และ (2)คอนโทรลเพิ่มเติม (Custom Control) คอนโทรลมาตรฐานเป็นคอนโทรลพื้นฐานที่อยู่ในกล่องเครื่องมือ (Toolbox) ของซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมประยุกต์ ตัวอย่างคอนโทรลมาตรฐานเช่น กรอบอักษร (Text Box) ปุ่มคำสั่ง (Command Button) และคอมโบบอกซ์ (Combo Box) ส่วนคอนโทรลเพิ่มเติมมีลักษณะเป็นแอคทีฟเอกซ์คอนโทรลซึ่งออกแบบมาให้มีฟังก์ชันการทำงานพิเศษ นอกเหนือจากฟังก์ชันพื้นฐานของคอนโทรลมาตรฐาน ตัวอย่างคอนโทรลเพิ่มเติมเช่น แถบสถานะ (Status Bar) คอนโทรลสื่อสาร (Communications) และแผนภูมิ (Chart) เป็นต้น

ผู้พัฒนาโปรแกรมประยุกต์ที่ต้องการให้โปรแกรมประยุกต์ใช้วินโดวส์คอนโทรลที่มีฟังก์ชันพิเศษนอกเหนือจากวินโดวส์คอนโทรลที่มีอยู่ในระบบ ผู้พัฒนานั้นสามารถสร้างแอคทีฟเอกซ์คอนโทรลที่มีฟังก์ชันพิเศษที่ต้องการขึ้นมาใช้งานเองได้ และเนื่องจากแอคทีฟเอกซ์คอนโทรลเป็นไลบรารีที่มีลักษณะเป็นดีแอลแอลแบบหนึ่งซึ่งใช้นามสกุล .OCX ดังนั้นซอฟต์แวร์ใดที่สามารถบรรจุแอคทีฟเอกซ์คอนโทรล (ActiveX Control Container) ได้ ก็สามารถเรียกใช้

---

[6] จิรพัฒน์ จันทร์เจิดศักดิ์ และวีระ นพนิราพาธ, เขียนโปรแกรมบน Microsoft Windows, บริษัทซีเอ็ดยูเคชั่น จำกัด, ห้างหุ้นส่วนจำกัด เอช-เอน การพิมพ์, 2521.

แอคทีฟเอกซ์คอนโทรลต่าง ๆ ที่ได้ลงทะเบียน (Register) ไว้กับระบบปฏิบัติการวินโดวส์มาใช้งานได้ ตัวอย่างซอฟต์แวร์ที่สามารถบรรจุแอคทีฟเอกซ์คอนโทรล เช่น ไมโครซอฟท์วิชวลเบสิก (Microsort Visual Basic) ไมโครซอฟท์แอคเซส (Microsoft Access) และอินเตอร์เนทบราวซ์เซอร์ (Internet Browser) เป็นต้น แอคทีฟเอกซ์คอนโทรลสามารถใช้ในซอฟต์แวร์ที่เป็นอินเตอร์เนทบราวซ์เซอร์ได้ เนื่องจากบราวซ์เซอร์สามารถบรรจุ (Download) คอนโทรลที่มีลักษณะเป็นดีแอลแอล และดำเนินการ (Execute) ในเนื้อที่โปรเซสของบราวซ์เซอร์ได้

แอคทีฟเอกซ์คอนโทรลได้เพิ่มส่วนที่เป็นส่วนต่อประสานกับโปรแกรมประยุกต์ไว้ให้เรียกใช้งานได้แก่ คุณสมบัติ (Property) วิธีการ (Method) และเหตุการณ์ (Event) ตัวอย่างคุณสมบัติของแถบสถานะ เช่น ชื่อ ความกว้าง และความยาวของแถบสถานะ ซึ่งสามารถกำหนดคุณสมบัติเหล่านี้ได้ในขณะออกแบบ (Design-Time) ตัวอย่างวิธีการหรือฟังก์ชันที่แถบสถานะเตรียมไว้ให้เรียกใช้ได้แก่ วิธีการเคลื่อนย้าย (Move Method) และวิธีการรีเฟรช (Refresh Method) ส่วนเหตุการณ์จะเป็นเหตุการณ์ที่เกิดขึ้นกับแอคทีฟเอกซ์คอนโทรล โดยที่ผู้พัฒนาสามารถเขียนคำสั่งให้โปรแกรมทำงานอย่างหนึ่ง เมื่อแอคทีฟเอกซ์คอนโทรลเกิดเหตุการณ์นี้ขึ้น ตัวอย่างเหตุการณ์ของแถบสถานะ เช่น ผู้ใช้คลิกเมาส์ที่แถบสถานะ (Click Event)

เครื่องมือที่พัฒนาขึ้นจะมีลักษณะเป็นแอคทีฟเอกซ์คอนโทรลที่ให้ผู้พัฒนาโปรแกรมประยุกต์ใด ๆ ที่ต้องการใช้ฟังก์ชันในการค้นหาข้อมูล สามารถเรียกใช้เครื่องมือนี้ได้ในขั้นตอนการพัฒนาโปรแกรมประยุกต์

2.4 ส่วนต่อประสานโอดีบีซี

ส่วนต่อประสานโอดีบีซี (ODBC: Open Database Connectivity) เป็นมาตรฐานบนระบบปฏิบัติการวินโดวส์ที่บริษัทไมโครซอฟท์เป็นผู้พัฒนาขึ้น ทำให้โปรแกรมประยุกต์สามารถเข้าถึงและประมวลผลในฐานข้อมูลบนระบบบริหารฐานข้อมูล (DataBase Management System) ต่าง ๆ ได้โดยใช้ภาษาสอบถามเชิงโครงสร้าง (SQL: Structured Query Language) เป็นภาษาในการเข้าถึงฐานข้อมูล จุดมุ่งหมายของโอดีบีซีคือ เพื่อที่จะให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ที่สามารถเข้าถึงฐานข้อมูลต่าง ๆ ที่อยู่บนระบบบริหารฐานข้อมูลที่แตกต่างกันได้ โดยไม่จำเป็นที่จะต้องทำการแปล (Compile) รหัสต้นฉบับใหม่

ส่วนต่อประสานโอดีบีซีได้ออกแบบมาในรูปแบบส่วนต่อประสานกับโปรแกรมประยุกต์ (API: Application Programming Interface) เพื่อให้โปรแกรมประยุกต์เรียกใช้ ทำให้สามารถเข้าถึง

ฐานข้อมูลบนระบบบริหารฐานข้อมูลใด ๆ ได้ โดยผู้พัฒนาจะต้องติดตั้งไดรฟเวอร์ (Driver) ของระบบบริหารฐานข้อมูลที่ต้องการใช้ลงไปก่อน โปรแกรมประยุกต์จึงจะสามารถเข้าถึงฐานข้อมูลที่ต้องการโดยใช้ส่วนต่อประสานโอดีบีซีได้

2.4.1 สถาปัตยกรรมโอดีบีซี

สถาปัตยกรรมของโอดีบีซี[7] (ODBC Architecture) ประกอบด้วย 4 ส่วน ดังนี้

1. โปรแกรมประยุกต์ จะทำการประมวลผลและเรียกใช้ฟังก์ชันของโอดีบีซีในการติดต่อ (Connect) กับฐานข้อมูลที่ต้องการ จากนั้นจะส่งข้อความสั่งในลักษณะภาษาสอบถามเชิงโครงสร้าง (SQL Statements) ไปยังระบบบริหารฐานข้อมูล จากนั้นนำผลลัพธ์ที่ได้กลับมา (Retrieve) และยกเลิกการติดต่อ (Disconnect) กับฐานข้อมูล

2. ตัวจัดการไดรฟเวอร์ (Driver Manager) มีลักษณะเป็นดีแอลแอล ที่จัดการในส่วนของการติดต่อสื่อสารระหว่างโปรแกรมประยุกต์กับไดรฟเวอร์ และทำหน้าที่เรียกใช้(Load) และปลดปล่อย (Unload) ไดรฟเวอร์ที่โปรแกรมประยุกต์ต้องการ รวมทั้งจะทำการประมวลผลฟังก์ชันโอดีบีซีบางฟังก์ชัน และบางฟังก์ชันจะส่งผ่านให้กับไดรฟเวอร์

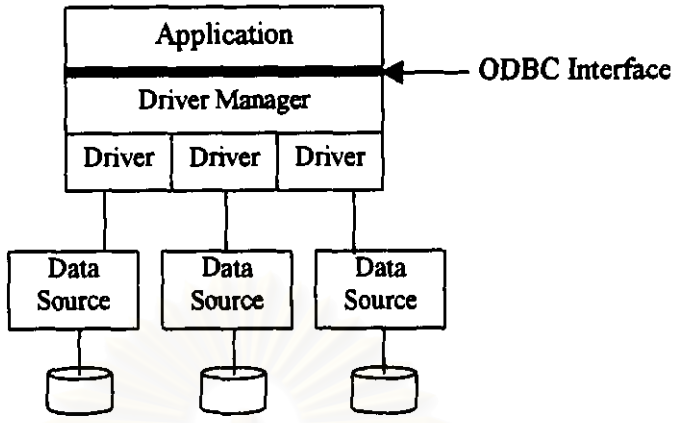
3. ไดรฟเวอร์ (Driver) เป็นดีแอลแอลที่ทำหน้าที่ในการประมวลผลฟังก์ชันต่าง ๆ ของโอดีบีซี โดยที่แต่ละไดรฟเวอร์ของระบบบริหารฐานข้อมูลจะแตกต่างกัน ไดรฟเวอร์จะทำหน้าที่ในการติดต่อกับดาต้าซอร์ส จากนั้นจะส่งคำร้องขอในรูปแบบภาษาสอบถามเชิงโครงสร้าง (SQL Request) ไปยังดาต้าซอร์สเพื่อให้ทำการประมวลผล ซึ่งในบางกรณีไดรฟเวอร์อาจจะต้องปรับปรุงคำร้องขอที่ได้จากโปรแกรมประยุกต์เพื่อให้คำร้องขอนั้นถูกต้องตามวากยสัมพันธ์ (Syntax) ของระบบบริหารฐานข้อมูลที่ระบุในดาต้าซอร์ส จากนั้นจะส่งค่าผลลัพธ์ (Result) ที่ได้กลับไปให้โปรแกรมประยุกต์ และยกเลิกการติดต่อกับดาต้าซอร์ส

4. ดาต้าซอร์ส (Data Source) ประกอบด้วยข้อมูลต่าง ๆ ที่ใช้ในการติดต่อและเข้าถึงฐานข้อมูล ได้แก่ ชื่อผู้ใช้ ไดรฟเวอร์ของระบบบริหารฐานข้อมูล รวมทั้งลักษณะของเครือข่าย (Network Platform)

---

[7] Microsoft Corporation, Microsoft ODBC 2.0 Programmer's Reference and SDK Guide: For Microsoft Windows and Windows NT, Microsoft Press, 1994.

ลักษณะของสถาปัตยกรรมโอดีบีซี แสดงได้ดังรูปที่ 2.3



รูปที่ 2.3 สถาปัตยกรรมโอดีบีซี [8]

เครื่องมือที่พัฒนาขึ้นจะใช้ส่วนต่อประสานโอดีบีซีในการติดต่อกับฐานข้อมูลที่ผู้ใช้ระบุ และจัดการเกี่ยวกับการรักษาความปลอดภัย (Security) ของข้อมูลในฐานข้อมูลสำหรับผู้ใช้แต่ละคน ส่งคำร้องขอต่าง ๆ ไปยังระบบบริหารฐานข้อมูล รวมทั้งส่งผลลัพธ์หรือข้อมูลที่ได้กลับมาให้กับเครื่องมือซึ่งเครื่องมือจะนำข้อมูลเหล่านั้นไปสร้างเป็นโครงสร้างข้อมูลแบบทู-ทรัย เพื่อที่จะให้ผู้ใช้สามารถค้นหาข้อมูลที่ต้องการ

เนื่องจากทฤษฎีบางทฤษฎีที่นำมาใช้ในงานวิจัยนี้จะต้องทำการปรับปรุงขั้นตอนวิธีบางอย่างเพื่อทำให้เครื่องมือที่พัฒนาขึ้นสามารถทำงานได้อย่างมีประสิทธิภาพดีขึ้น ทฤษฎีที่จะทำการปรับปรุงได้แก่ทฤษฎีของโครงสร้างข้อมูลแบบทู-ทรัยที่จะนำมาใช้เป็นโครงสร้างข้อมูลของดัชนี ซึ่งรายละเอียดในการปรับปรุงจะกล่าวไว้ในบทถัดไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

[8] Ibid., p. 5.