บทที่ 5

แนวทางการพัฒนา

5.1 แนวทางการพัฒนา

ลักษณะของโมเดลข้อมูลแบบไดเร็กเตดรีเคอร์ซีฟเลเบลโหนดไฮเพอร์กราฟ มีส่วนที่สำคัญคืออะตอมมิคเลเบลโหนด เลเบลโหนดเชิงซ้อน ไฮเพอร์กราฟ และเลเบลโหนดเชิงซ้อนที่มีเลเบลโหนดเชื่อมต่อ จากส่วนประกอบต่าง ๆ ของโมเดลข้อมูลนี้ สามารถนำไปประยุกต์ใช้งานจริงได้ โดยในขั้นแรกต้องกำหนดโครงสร้างข้อมูลให้สอดคล้องกับโมเดลข้อมูลนี้ และให้เอื้ออำนวยต่อการเชื่อมโยงกับผู้ใช้ เพื่อเป็นแนวทางสู่การพัฒนาต่อไป ผู้วิจัยได้ออกแบบโครงสร้างข้อมูลต่าง ๆ โดยมีรายละเอียดดังนี้

5.1.1 โครงสร้างข้อมูลหน่วยข้อมูล

หน่วยข้อมูลเป็นหน่วยเก็บเนื้อความของเลเบลโหนด โดยมีโครงสร้างข้อมูลแสดงด้วยภาษาซีได้ดังนี้

```
struct dataunit{
        integer   datatype;
        union{
            char  *data1;
           /* ข้อมูลประเภทอื่น ๆ */
            ...
            } udata;
        struct dataunit *nextdata;
        }
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากโครงสร้างข้อมูลนี้ udata เป็นโครงสร้างที่ระบุด้วย union โดยใช้ datatype เป็นตัวบ่งบอกประเภทของข้อมูลภายใน union ว่าเป็นประเภทใด การกำหนดค่าให้กับ datatype ขึ้นอยู่กับผู้พัฒนา เช่น ถ้า datatype = 1 แสดงว่าเนื้อหาเป็นประเภทข้อความที่แทนด้วยตัวแปร data1 เป็นต้น

5.1.2 โครงสร้างข้อมูลของเลเบลโหนด

โครงสร้างข้อมูลของเลเบลโหนดเป็นโครงสร้างพื้นฐานอีกโครงสร้างหนึ่งที่มีความสำคัญ โดยแต่ละเลเบลโหนดจะมีส่วนประกอบต่าง ๆ ดังนี้

- id เป็นเลขประจำของเลเบลโหนด
- name เป็นชื่อของเลเบลโหนด
- parent เป็นเซตตัวชี้ที่ชี้ไปยังเลเบลโหนดที่เลเบลโหนดนั้นเป็นสมาชิก
- cur_p เป็นตัวชี้ที่ชี้ไปยังโหนดปัจจุบันใน parent เพื่อบ่งบอกว่าเลเบลโหนดนี้กำลังเป็นสมาชิกของเลเบลโหนดใดอยู่ ในกรณีที่เลเบลโหนดถูกใช้ร่วมกัน
- member เป็นเซตของตัวชี้ที่ชี้ไปยังเลเบลโหนด หรือไฮเพอร์อาร์คที่เป็นสมาชิกของเลเบลโหนดนั้น ๆ
- cur_m เป็นตัวชี้ที่ชี้ไปยังโหนดปัจจุบันใน member
- linkto เป็นเซตของตัวชี้ที่ชี้ไปยังเลเบลโหนดอื่น ๆที่เลเบลโหนดนั้น ๆ อ้างถึง
- cur_l เป็นตัวชี้ที่ชี้ไปยังโหนดปัจจุบันใน linkto
- info เป็นตัวชี้ที่ชี้ไปยังหน่วยข้อมูล
- typenode เป็นตัวบ่งบอกประเภทของเลเบลโหนด ถ้า typenode = 0 แล้ว เป็นอะตอมมิคเลเบลโหนด ถ้า typenode = 1 แล้ว เป็นเลเบลโหนดเชิงซ้อน
- nextln เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนดตัวต่อไป

จากส่วนประกอบต่าง ๆสามารถเขียนโครงสร้างข้อมูลด้วยภาษาซีได้ดังนี้

```
type  struct node{
        integer id;      /* หมายเลขเลเบลโหนด */
        integer typenode;    /* ประเภทของเลเบลโหนด*/
        char    *name;  /* ชื่อของเลเบลโหนด */
        struct  parent_node{
                integer  depend;
                struct node *parentnode;
                struct parent_node *prevparent;
                struct parent_node *nextparent;
        } *parent; /*เซตของเลเบลโหนดที่เลเบลโหนดนี้เป็นสมาชิก*/
        struct parent_node *cur_p; /*ตัวชี้ที่ชี้ไปยัง parent ปัจจุบัน*/
        struct memnode {
                integer depend;
                integer tag;
                union{
                        struct hyperarcnode{
                           integer id_hp;
                           char    *name_hp;
                           struct node *parent_node;
                           struct hyperarcnode *cur_arc;
                           integer  tag;
                           union {
                                struct hyperarcnode *arc;
                                struct node    *memnode;
                           } hyperarcu;
                        struct hyperacrnode *hyperarcnext;
                } *memarc;
```

```
                struct node *memnode;
            } memptr;
            struct  memnode *memprev;
            struct  memnode *memnext;
        } *member;  /* ตัวชี้ที่ชี้ไปยังเซตของสมาชิกโหนดนี้ */
        struct memnode  *cur_m;
        struct dataunit *info; /*ตัวชี้ที่ชี้ไปยังหน่วยข้อมูล*/
        struct  linknode {
                        struct node *target;
                        struct hyperarcnode *memnode;
                        struct hyperacrnode *hyperarclink;
                        struct linknode *linkprev;
                        struct linknode *linknext;
        } *linkto; /*ตัวชี้ที่ชี้ไปยังเลเบลโหนด/ไฮเพอร์อาร์ค*/
        struct  node *nextln;
} labelnode;
```

จากโครงสร้างข้อมูลที่เขียนด้วยภาษาซี สามารถเขียนแทนด้วยรูปภาพได้ดังรูปที่ 5.1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

labelnode

id	name	parent	cur_p	member	cur_m	info	linkto	cur_l	nextln

parent

depend	parentnode	prevparent	nextparent

member

depend	tag	memptr	memprev	memnext

memptr

memarc/memnode

linkto

target	memnode	hyperarclink	linkprev	linknext

hyperarclink

id_hp	name_hp	parent	cur_arc	tag	hyperarcu	hyperarnext

hyperarcu

arc/memnode

รูปที่ 5.1 แผนภาพแสดงโครงสร้างข้อมูลของเลเบลโหนด

จากรูปที่ 5.1 โครงสร้างข้อมูลของเลเบลโหนด ในส่วนของ parent member และ linkto โดยแต่ละส่วนจะมีรายละเอียดดังนี้

- parent เป็นเซตของตัวชี้ที่ชี้ไปยังเลเบลโหนดที่เลเบลโหนดนั้น ๆ เป็นสมาชิก โดยที่ parent ประกอบด้วย parentnode(เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนดที่เลเบลโหนดนั้น ๆเป็นสมาชิก) prevparent(เป็นตัวชี้ที่ชี้ไปยังโหนดก่อนหน้าของโหนดปัจจุบันที่ชี้โดย cur_parent) nextparent(เป็นตัวชี้ไปยังโหนดตัวต่อไปของ parent) และ depend (เป็นตัวบ่งบอกว่าเลเบลโหนดนี้ขึ้นอยู่กับ parent หรือไม่ ถ้า depend = 0 แสดงว่าไม่ขึ้นอยู่กับ parent ถ้า depend = 1 แล้ว เลเบลโหนดนี้จะขึ้นอยู่กับ parent) การใช้ depend จะมีประโยชน์ในการตรวจสอบการเป็นสมาชิกของเลเบลโหนดนี้ในกรณีที่มีการลบเลเบลโหนดนี้

การระบุให้ parentเป็นเซตก็เพราะว่า เลเบลโหนดหนึ่ง ๆสามารถถูกใช้ร่วมกันได้ในหลาย ๆ เลเบลโหนด เช่นภาพหนึ่งภาพอาจอยู่ได้ในหลาย ๆ ส่วนของเอกสาร เป็นต้น

- member เป็นเซตของตัวชี้ที่ชี้ไปยังเลเบลโหนดหรือไฮเพอร์อาร์คที่เป็นสมาชิกของเลเบลโหนดนี้ ซึ่งส่วนประกอบของแต่ละโหนดใน member คือ depend tag memptr memprev และ memnext โดยแต่ละส่วนจะมีรายละเอียดดังนี้

▿ depend เป็นตัวบ่งบอกว่าสมาชิกนั้น ขึ้นอยู่กับเลเบลโหนดนี้หรือไม่ กำหนดให้ ถ้า depend = 0 แล้ว สมาชิกนี้ไม่ขึ้นอยู่กับเลเบลโหนดนี้ แต่ถ้า depend = 1 แล้ว สมาชิกนี้ขึ้นอยู่กับเลเบลโหนดนี้ การกำหนดให้ depend ใน member จะมีประโยชน์ในการตรวจสอบว่าสมาชิกใดบ้างที่มีการเป็นสมาชิกที่ขึ้นอยู่กับparentใด ในกรณีที่ต้องลบเลเบลโหนดนี้

▿ tag เป็นตัวบ่งบอกว่าค่าใน memptr เป็น memarc หรือ memnode ถ้า tag = 0 ค่าใน memptr คือ memarc ซึ่งเป็นค่าตัวชี้ที่ชี้ไปยังไฮเพอร์อาร์คที่เป็นสมาชิกของเลเบลโหนดนี้ แต่ถ้า tag = 1 ค่าใน memptr คือ memnode ซึ่งเป็นค่าตัวชี้ที่ชี้ไปยังเลเบลโหนดที่เป็นสมาชิกของเลเบลโหนดนี้

▿ memprev เป็นตัวชี้ที่ชี้ไปยังโหนดก่อนหน้าโหนดปัจจุบันใน member

▽ memnext เป็นตัวชี้ที่ชี้ไปยังโหนดต่อไปของ member

- linkto เป็นเซตของตัวชี้ที่ชี้ไปยังเลเบลโหนดหรือไฮเพอร์อาร์คที่เลเบลโหนดนี้อ้างถึง และชี้ไปยังไฮเพอร์อาร์คที่เลเบลโหนดนี้สัมพันธ์ โดยประกอบด้วยส่วนต่าง ๆคือ target memnode hyperarclink linkprev และlinknext โดยแต่ละส่วนมีความหมายดังนี้

▽ target เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนดที่ต้องการอ้างถึง ใช้ในกรณีที่มีการชี้ไปยังสมาชิกของเลเบลโหนดเชิงซ้อนจำเป็นต้องระบุค่าตัวชี้นี้เสมอ

▽ memnode เป็นตัวชี้ที่ชี้ไปยังสมาชิกภายในไฮเพอร์อาร์ค (ที่ชี้โดย hyperarclink)

▽ hyperarclink เป็นตัวชี้ที่ชี้ไปยังไฮเพอร์อาร์ค การระบุ hyperarclink เข้าไปร่วมอยู่ใน linkto ก็เพื่อให้สามารถบ่งบอกได้ว่าเลเบลโหนดที่อ้างถึงหรือไฮเพอร์อาร์คที่อ้างถึงอยู่ภายใต้ไฮเพอร์อาร์คเส้นเดียวกันกับเลเบลโหนดปัจจุบัน ตัวอย่างเช่น ระบบไฮเพอร์เท็กซ์ ประกอบด้วย เลเบลโหนด a b c d โดยมีการเชื่อมโยงระบุโดยไฮเพอร์อาร์ค (a b) และ (a c d) จากทั้งสองไฮเพอร์อาร์คสมมติว่าเลเบลโหนดปัจจุบันอยู่ที่ a เส้นทางที่เลเบลโหนด a จะไปได้มีอยู่สองเส้นทาง คือจาก a ไป b และ จาก a ไป c ฉะนั้นจำนวนโหนด ของ linkto ของ a จะมีสองโหนดคือ โหนดที่บรรจุค่าตัวชี้ที่ไปยังโหนด (ในไฮเพอร์อาร์ค (a b) ที่ชี้โดย hyperarclink) ที่บรรจุค่าตัวชี้ที่ชี้เลเบลโหนด b ส่วนอีก โหนดหนึ่งจะบรรจุค่าตัวชี้ที่ชี้ไปยังโหนด(ในไฮเพอร์อาร์ค (a c d) ที่ชี้โดย hyperarclink) ที่บรรจุค่าตัวชี้ที่ชี้ไปยังเลเบลโหนด c จากลักษณะการออกแบบเช่นนี้ ทำให้สะดวกในการอ้างถึง และช่วยให้สามารถพัฒนาวิธีการสำรวจเส้นทางล่วงหน้าได้ เป็นต้น

5.1.3 โครงสร้างข้อมูลของไฮเพอร์อาร์ค

การระบุการเข้าถึงของเลเบลโหนดในลักษณะแบบตามลำดับในโมเดลข้อมูลนี้จะใช้ไฮเพอร์อาร์คเป็นตัวระบุความสัมพันธ์ดังกล่าว จากการพิจารณาลักษณะของไฮเพอร์เท็กซ์จะเห็นว่าโครงสร้างข้อมูลที่เหมาะสมสำหรับไฮเพอร์อาร์คมากที่สุดคือโครงสร้างข้อมูลแบบลิส ตัวอย่างโครงสร้างข้อมูลแบบลิสคือ L1 = (5, 6, 7, 2) เป็นลิสของตัวเลข 5 6 7 2 ที่มีชื่อว่า L1 และ L2= ((2,3,4),5) เป็นลิสที่ประกอบด้วยสมาชิกคือลิส (2,3,4) และ 5 เป็นต้น (Tenenbaum, Langsam, Augenstein, 1990)

การนำโครงสร้างข้อมูลแบบลิสแทนไฮเพอร์อาร์คนั้น โหนดแต่ละโหนดจะประกอบด้วยส่วนประกอบอย่างน้อย 3 ส่วนคือ tag memarc/memnode nodenext โดยสามารถแสดงได้ดังรูปที่ 5.2

tag	memarc/memnode	nodenext

รูปที่ 5.2 แสดงส่วนประกอบของโหนดในโครงสร้างข้อมูลแบบลิส

จากรูปที่ 5.2 ค่าของ tag จะเป็นตัวบ่งบอกว่าโหนดใดเป็นลิสหรือเป็นเพียงโหนดธรรมดา ในที่นี้กำหนดให้ tag = 0 ให้หมายความว่าโหนดนั้นเป็นลิส ถ้า tag = 1 ให้หมายความว่าโหนดนั้นเป็นโหนดธรรมดา ค่าของ memarc/memnode เป็นตัวชี้ที่ชี้ไปยังลิสหรือโหนดอื่น ๆ อย่างใดอย่างหนึ่ง ค่าของ nodenext เป็นตัวชี้ที่ชี้ไปยังโหนดต่อไปของลิสนี้

จากโครงสร้างของลิสสามารถนำมาประยุกต์ใช้กับโครงสร้างข้อมูลของไฮเพอร์อาร์คได้ดังในรูป 5.1 โครงสร้างข้อมูลของแต่ละไฮเพอร์อาร์คประกอบด้วย id_hp name_hp parent_node cur_arc tag hyperarcu hyperarcnext โดยแต่ละส่วนจะมีความหมายดังนี้

- id_hp เป็นเลขประจำไฮเพอร์อาร์ค

- name_hp เป็นชื่อของไฮเพอร์อาร์ค

- parent_node เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนดที่เป็น parent ของเลเบลโหนด ที่ชี้โดยฟิลด์ memnode ของโหนดนี้

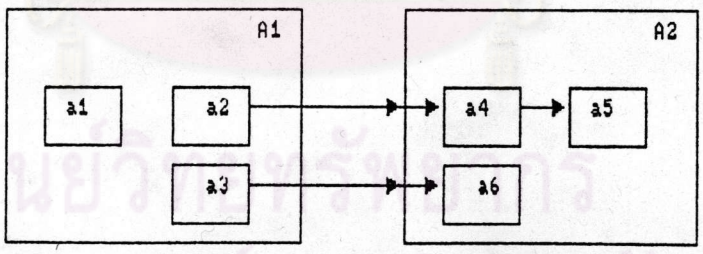
- cur_arc เป็นตัวชี้ที่ชี้ไปยังโหนดปัจจุบันที่อยู่ภายใต้ไฮเพอร์อาร์คเส้นนั้น เพื่อช่วยให้ทราบว่าขณะนี้กำลังทำงานอยู่ในระดับใด

- tag เป็นตัวบ่งบอกว่า สมาชิกภายในไฮเพอร์อาร์คเป็นเลเบลโหนด หรือไฮเพอร์อาร์ค ถ้า tag = 0 แล้ว ค่าของ hyperarcu เป็น arc ซึ่งเป็นตัวชี้ที่ชี้ไปยังไฮเพอร์อาร์ค แต่ถ้า tag =1 ค่าของ hyperarcu เป็น memnode ซึ่งเป็นตัวชี้ที่ชี้ไปยังเลเบลโหนด

- hyperarcnext เป็นตัวชี้ที่ชี้ไปยังโหนดต่อไปที่เป็นไฮเพอร์อาร์ค

จากโครงสร้างข้อมูลต่าง ๆ สามารถนำไปใช้งานได้และสอดคล้องกับโมเดลข้อมูลแบบไดเร็กเตดรีเคอร์ชีฟเลเบลโหนดไฮเพอร์กราฟดังตัวอย่างที่ 5.1

ตัวอย่าง 5.1 ตัวอย่างโครงสร้างของเครือข่ายภายใต้โมเดลข้อมูลแบบไดเร็กเตดรีเคอร์ชีฟเลเบลโหนดไฮเพอร์กราฟ แสดงได้ดังรูปที่ 5.3



รูปที่ 5.3 เครือข่ายแสดงการเชื่อมโยงระหว่างเลเบลโหนด

ในรูปที่ 5.3 เครือข่ายประกอบด้วยเลเบลโหนดเชิงซ้อนคือ A1 และ A2 โดยที่สมาชิกของ A1 คืออะตอมมิคเลเบลโหนด a1 a2 และ a3 โดยที่การเป็นสมาชิกของ a1 a2 และ a3 ไม่ขึ้นกับ A1 ส่วน A2ประกอบด้วยสมาชิกคือไฮเพอร์อาร์ค (a4 a5) และอะตอมมิคเลเบลโหนด a6 โดยแทนเซตของเลเบลโหนดแสดงได้ดังในรูปที่ 5.4 จากรูปที่ 5.4 ค่า headln เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนด

แรกเสมอ ส่วนค่า currentln เป็นตัวชี้ที่ชี้ไปยังเลเบลโหนดปัจจุบัน ซึ่งค่าของตัวชี้นี้จะแปรเปลี่ยนไปตามการอ้างอิงในระหว่างการใช้งาน สำหรับการเชื่อมโยงระหว่างเลเบลโหนดในรูป การเชื่อมโยงจาก A1 ไปยัง A2 มีสองจุดคือผ่านจาก a2 ไปยังไฮเพอร์อาร์ค (a4 a5) และอีกเส้นทางหนึ่งคือจาก a3 ไป a6 จากเครือข่ายของเลเบลโหนดดังกล่าวนี้เราสามารถเขียนแทนเซตของไฮเพอร์อาร์คได้ดังรูปที่ 5.5

จากรูปที่ 5.5 headhp เป็นตัวชี้ที่ชี้ไปยังโหนดแรกของไฮเพอร์อาร์คเสมอ และ currenthp เป็นตัวชี้ที่ชี้ไปยังไฮเพอร์อาร์คปัจจุบัน จากการอ้างอิงระหว่างเลเบลโหนด A1 ไปยัง A2 แสดงได้ดังรูปที่ 5.6 โดยแสดงเฉพาะเลเบลโหนด A1 A2 และ a2

จากรูปที่ 5.6 ตัวอักษรที่แทนเลเบลโหนดในที่นี้คือตัวชี้ที่ชี้ไปยังเลเบลโหนดนั่นเอง เลเบลโหนด A1 ประกอบด้วยสมาชิกคือ a1 a2 และ a3 และมีการเชื่อมโยงไปยัง A2 ส่วนอะตอมมิคเลเบลโหนด a2 จะมีการเชื่อมโยงไปไฮเพอร์อาร์ค (a4 a5) ที่เป็นสมาชิกใน A2 ฉะนั้นการระบุการอ้างอิงต้องระบุ parent ของสมาชิกที่อ้างถึงด้วย เป็นต้น

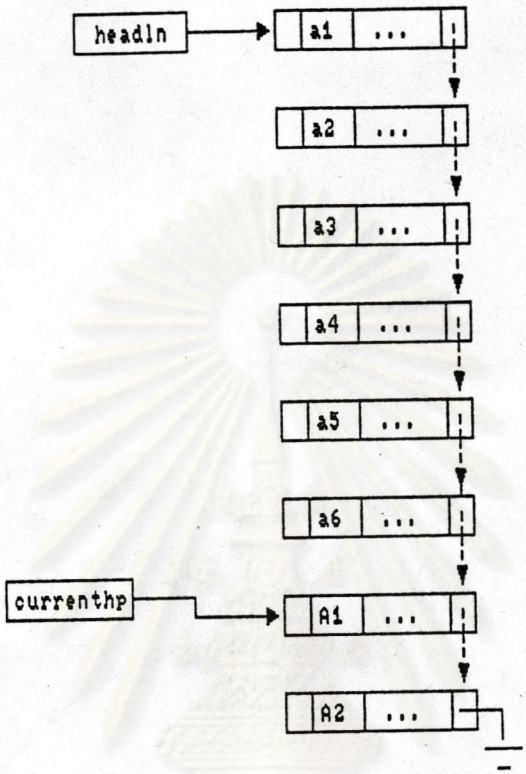
5.2 การตอบสนองต่อโอเปอร์เรชันการเบราส์ของโครงสร้างข้อมูลของเลเบลโหนดและโครงสร้างข้อมูลของไฮเพอร์อาร์ค

การออกแบบโครงสร้างข้อมูลของเลเบลโหนดและโครงสร้างข้อมูลของไฮเพอร์อาร์ค ผู้วิจัยได้พยายามออกแบบให้สอดคล้องกับโมเดลข้อมูลแบบไดเร็กเตดรีเคอร์ซีฟเลเบลโหนดไฮเพอร์กราฟให้มากที่สุด พร้อมทั้งพิจารณาว่าโครงสร้างข้อมูลต่าง ๆเหล่านี้ สามารถตอบสนองต่อโอเปอร์เรชันการเบราส์ต่าง ๆ เช่นตัวอย่างดังต่อไปนี้

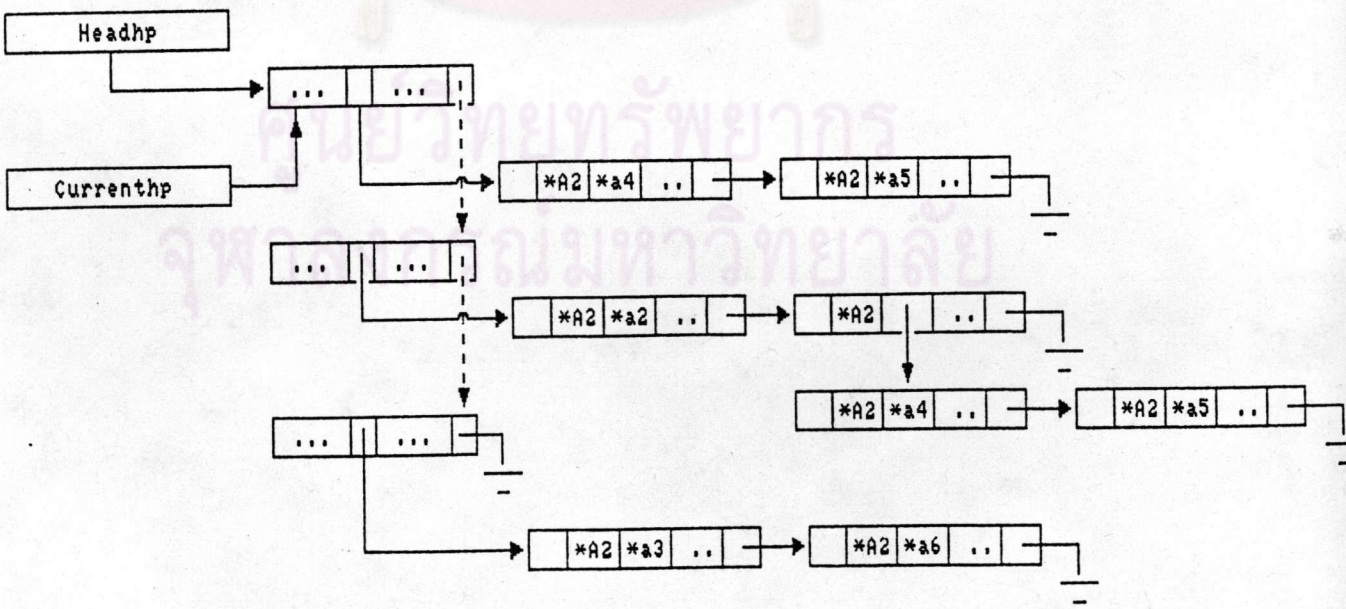
(1) การตอบสนองต่อการเบราส์แบบซูมเข้าและซูมออก

จากโครงสร้างข้อมูลของเลเบลโหนด ผู้วิจัยได้พยายามออกแบบให้อำนวยประโยชน์ต่อการพัฒนาการเบราส์แบบซูม โดยเฉพาะในส่วนของ member สมาชิกของ member คือตัวชี้ที่ชี้ไปยังเลเบลโหนดหรือไฮเพอร์อาร์ค

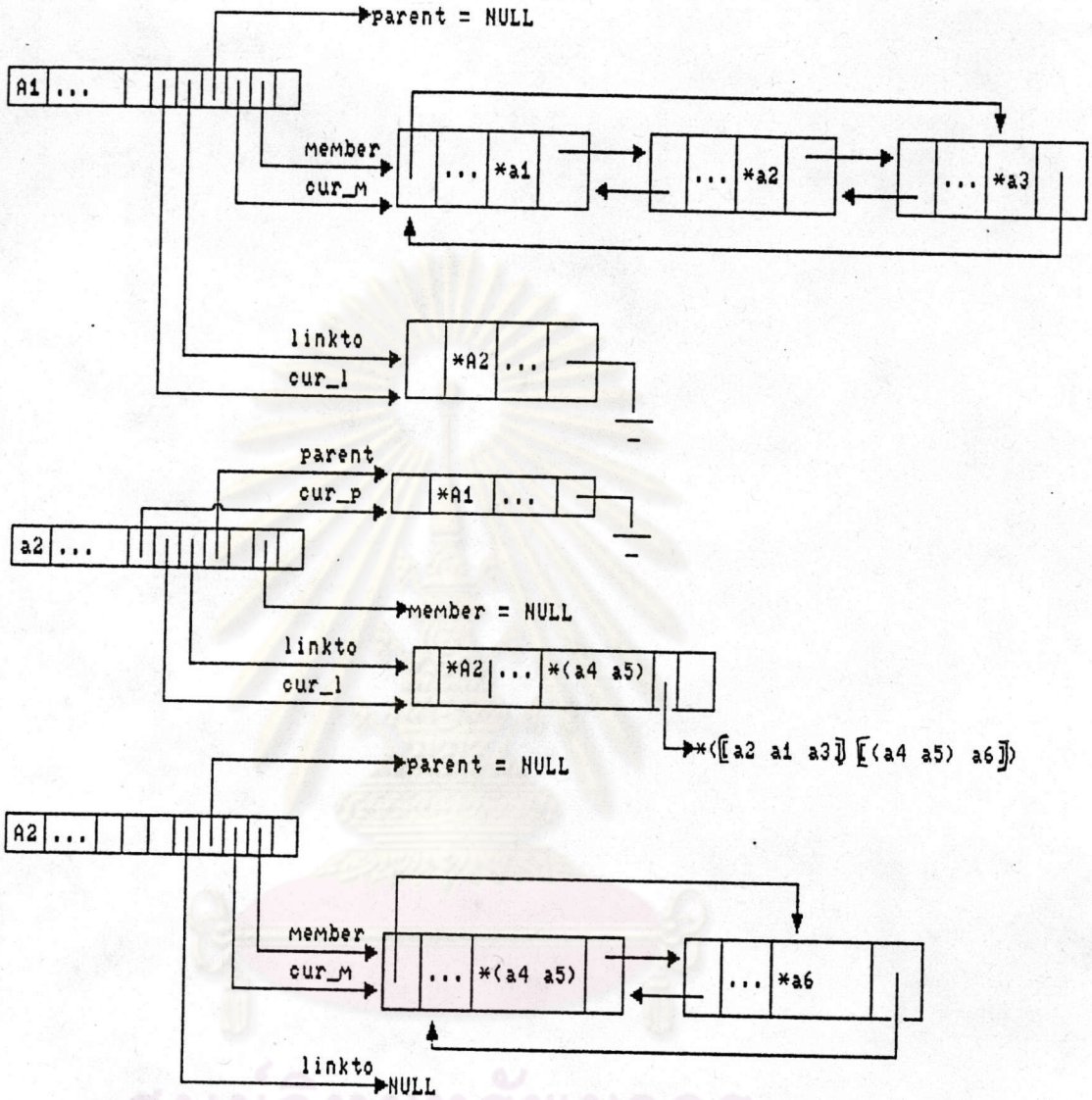
ในกรณีที่ชี้ไปยังเลเบลโหนดเชิงซ้อนอื่น ๆ ผู้ใช้ก็สามารถเบราส์แบบซูมเข้า เพื่อเข้าไปอ่านสารสนเทศของสมาชิกของเลเบลโหนดนั้น ในขณะเดียวกันผู้ใช้ก็สามารถเบราส์แบบซูมออก เพื่อออกไปยังระดับบนได้



รูปที่ 5.4 แสดงเซตของเลเบลโหนด ในตัวอย่างที่ 5.1



รูปที่ 5.5 แสดงเซตของไฮเพอร์อาร์ค ในตัวอย่างที่ 5.1



รูปที่ 5.6 แสดงโหนดต่าง ๆ ภายในเครือข่ายของเลเบลโหนดในตัวอย่าง 5.1 โดยกำหนดค่าตัวชี้ด้วย * แล้วตามด้วยชื่อเลเบลโหนด หรือไฮเพอร์อาร์คนั้น ๆ

(2) การตอบสนองต่อการเบราส์แบบเพน

ดังในตัวอย่างที่ 1 a2 มีการอ้างอิงไปยังไฮเพอร์อาร์ค (a4 a5) ซึ่งเป็นสมาชิกของเลเบลโหนดเชิงซ้อน A2 หลังจากที่มีการอ่านสารสนเทศของเลเบลโหนด a4 และ a5 แล้ว ผู้ใช้ต้องการอ่านเลเบลโหนด a6 ในลักษณะเช่นนี้โอเปอร์เรชันที่เหมาะสมก็คือโอเปอร์เรชันเพนนั่นเอง เมื่อพิจารณา

จากโครงสร้างข้อมูลที่ออกแบบ จะพบว่าโครงสร้างข้อมูลต่าง ๆนั้นได้สนับสนุนการทำงานของโอเปอร์เรชันนี้ได้เป็นอย่างดี

(3) การตอบสนองต่อการเบราส์แบบจวบกัน

ดังตัวอย่างที่ 1 สมาชิกของ A1 คือ เลเบลโหนด a1 a2 และ a3 จากโครงสร้างข้อมูลของเลเบลโหนด A1 สมาชิกที่ member ชี้คือ ตัวชี้ที่ชี้ไปยังเลเบลโหนด a1 a2 และ a3 ในทางปฏิบัติแล้วอาจแทนตัวชี้ต่าง ๆเหล่านี้อยู่ในรูปของสัญรูปใด ๆ เพื่อบ่งบอกการเชื่อมโยงได้ ผู้ใช้เพียงแต่เลือกปุ่มใด ๆ หรือทั้งหมดก็ได้ ก็จะทำให้ผู้ใช้สามารถเข้าถึงเลเบลโหนดต่าง ๆเหล่านี้ได้พร้อมกัน เป็นต้น

(4) การเบราส์แบบเชิงเส้นโดยอัตโนมัติ

โครงสร้างข้อมูลของเลเบลโหนด สมาชิกใน member ที่เป็นไฮเพอร์อาร์คสามารถกำหนด การเข้าถึงแบบตามลำดับโดยอัตโนมัติ โดยโอเปอร์เรชัน AUTO_SEQUENCE จากโครงสร้างของข้อมูลได้กำหนดให้ tag เป็นตัวบ่งชี้ว่าสมาชิกใดเป็นไฮเพอร์อาร์คหรือเลเบลโหนด ทำให้การพัฒนาโอเปอร์เรชันนี้ทำได้สะดวกยิ่งขึ้น

(5) การตอบสนองต่อโอเปอร์เรชัน SELECT_STEP

โครงสร้างข้อมูลของเลเบลโหนด เลเบลโหนดแต่ละเลเบลโหนดจะมีการเชื่อมโยงไปยังไฮเพอร์อาร์ค ที่เลเบลโหนดนั้นเป็นสมาชิก โดยโดยระบุลงในส่วนของ hyperarclink จากการกำหนดโครงสร้างข้อมูลเช่นนี้ ได้สนับสนุนการตรวจสอบว่าเลเบลโหนดปัจจุบันนี้อยู่ส่วนใดของเส้นทางนั้นหรืออยู่ลำดับที่เท่าใดภายใต้ไฮเพอร์อาร์คเส้นนั้น เมื่อทราบเช่นนี้ก็จะช่วยให้การพัฒนาโอเปอร์เรชัน SELECT_STEP ซึ่งเป็นโอเปอร์เรชันที่ให้ผู้ใช้เลือกการเบราส์แบบย้อนกลับหรือไปข้างหน้าตามเส้นทางนั้น ได้สะดวกยิ่งขึ้น เป็นต้น

จากโครงสร้างข้อมูลต่าง ๆ ภายใต้โมเดลข้อมูลนี้ ผู้วิจัยได้พยายามออกแบบให้มีแนวทางเป็นไปได้มากที่สุด เพื่อให้ผู้พัฒนาสามารถมีจินตนาการ และมีแนวทางในการนำเอาโมเดลข้อมูลนี้ไปพัฒนาต่อไป