

บทที่ 3

วิธีดำเนินการวิจัย

การวิจัยเรื่องการพัฒนาภาษาไพธอนสำหรับเขียนแบบก่อสร้างชิ้นส่วนโครงสร้างคอนกรีตเสริมเหล็ก เป็นการวิจัยเชิงประยุกต์ โดยมี ภาษา SVG และภาษา Python เป็นหน่วยการวิเคราะห์ และเพื่อให้การวิจัยครั้งนี้บรรลุตามวัตถุประสงค์ที่ตั้งไว้ ผู้วิจัยจึงได้กำหนดวิธีดำเนินการวิจัย ดังนี้

1. ศึกษารูปแบบคำสั่งของภาษา SVG

ในการสร้างคำสั่ง SVG สามารถใช้งานโปรแกรมใดก็ได้ที่มีฟังก์ชันสร้างข้อความ ในขั้นต้นได้ทำการทดสอบโดยเลือกใช้ภาษา Basic ของโปรแกรม OpenOffice แต่เมื่อทำการทดสอบและพิจารณาผลการดำเนินงานนำไปเปรียบเทียบกับภาษาไพธอน พบว่าภาษาไพธอนมีความยืดหยุ่นสูงกว่าในการพัฒนาคำสั่ง และเป็นภาษาที่ใช้ในการเขียนส่วนเสริม (Extension) ของโปรแกรม Inkscape ด้วย

```
REM ***** BASIC *****
" chakkree tiyawongsuwan
" start 10 Apr 2010
Sub Main
  'print SVG_Text(10,10,"Test")
  print SVG_Line(10 , 20 , 15 , 50)
End Sub

function SVG_Line(x1 , y1 , x2 , y2) as string
  str1 = "<line "
  str1 = str1 & " x1=" & chr(34) & x1 & chr(34)
  str1 = str1 & " y1=" & chr(34) & y1 & chr(34)
  str1 = str1 & " x2=" & chr(34) & x2 & chr(34)
  str1 = str1 & " y2=" & chr(34) & y2 & chr(34)
  str1 = str1 & "/>"
  SVG_Line = str1
end function

function SVG_Rectangle(x1 , y1 , x2 , y2) as string
  SVG_Rectangle = "<rect />"
end function

function SVG_Circle(x1 , y1 , radius) as string
  SVG_Circle = "<circle />"
end function

function SVG_Text(x1 , y1 , TextInput , optional FontSize ) as string
  if isMissing(FontSize) then FontSize=16
  str1 = "<text>"
  str1 = str1 & TextInput
  str1 = str1 & "</text>"
  'print str1
  SVG_Text = str1
end function
```

2. ศึกษาการใช้งานส่วนเขียน SVG ด้วยภาษาไพธอน

ในการใช้งานภาษาไพธอนเพื่อเขียน SVG นั้น มีผู้พัฒนาไว้หลายส่วน ที่มวิจัยได้ทำการศึกษาพบว่า แพคเกจ SVGFig มีความเหมาะสมที่สุด โดยที่ผู้พัฒนา SVGFig มีความประสงค์ใช้งานภาษา SVG ในการแสดงแผนภาพหรือกราฟ ที่สามารถคำนวณจากสมการได้ จะมีคลาสที่ชื่อว่า SVG สำหรับใช้ในการจัดการภาษา SVG

SVG

+t : string (SVG type)
+sub[] : SVG # list of element
+attr[] : dict # เป็นตัวแปร dictionary เก็บค่าคุณลักษณะ

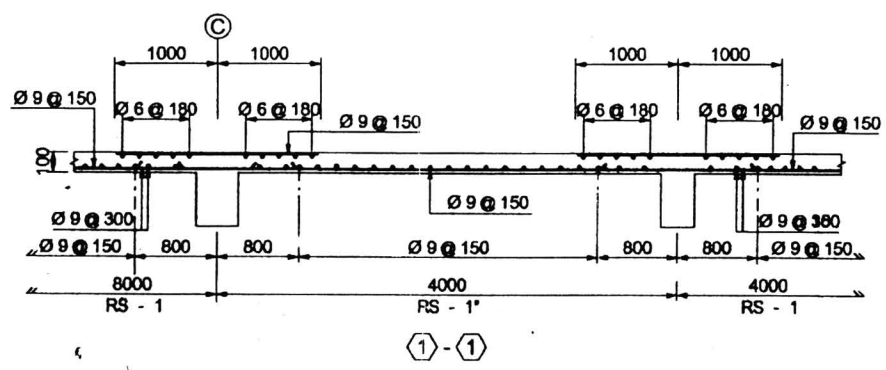
- __init__(*t_sub, **attr)
- __repr__()
- __str__()
- __getitem__ () # ในภาษาไพธอนทำให้ใช้งานวัตถุนี้แบบ list
- __setitem__ () # ในภาษาไพธอนทำให้ใช้งานวัตถุนี้แบบ list
- __delitem__ ()
- __contains__()
- __eq__() # โอเวอร์โหลดโอเปอเรเตอร์ ==
- __ne__() # โอเวอร์โหลดโอเปอเรเตอร์ <=
+append(x)
+prepend(x)
+extend(x)
+clone(shallow=False) # ใช้ในการสำเนาวัตถุ
+depth_first(depth_limit=None)
+breadth_first(depth_limit=None)
- __iter__()
+items()
+keys()
+values()
+tree()
+xml()
+standalone_xml()
- __standalone_xml()
+interpret_fileName()
+save(fileName=None, encoding="utf-8", compresslevel=None)
+inkview() # ใช้ได้ใน Linux ใน Windows ใช้ไม่ได้
+inkscape() # ใช้ได้ใน Linux ใน Windows ใช้ไม่ได้
+firefox() # ใช้ได้ใน Linux ใน Windows ใช้ไม่ได้

SVGDepthIterator

ภาพที่ 12 แผนภาพคลาสโคอะแกรมของ SVGFig

3. ขั้นตอนการทำงานของการพัฒนาโมดูลเขียนรายละเอียดแบบก่อสร้าง

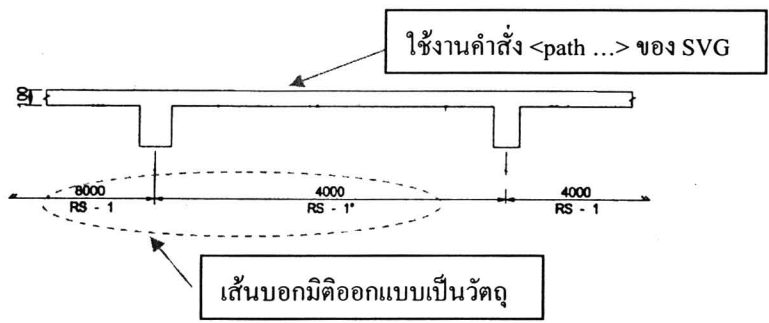
ขั้นตอนที่ 1 เลือกงานเขียนแบบที่เราต้องการนำมาสร้างสกริปต์ พิจารณาหาตัวแบบต่าง ๆ เช่น ความยาวความหนาของแผ่นพื้น



ภาพที่ 13 แบบรายละเอียดการเสริมเหล็กในแผ่นพื้น

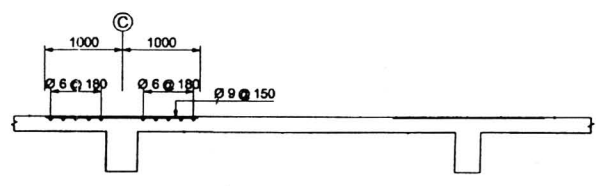
ขั้นตอนที่ 2 ทำการแยกส่วนของภาพให้เป็นแต่ละส่วน เพื่อให้เขียนคำสั่งได้สะดวกขึ้น โดยส่วนใหญ่ มักจะเป็น เส้นกรอบ เหล็กเสริมต่าง ๆ แผ่นพื้นก็นำมาแสดงเป็นตัวอย่างก็จะสามารถแบ่งได้ ดังนี้

ก. เส้นขอบเขต และเส้นบอกมิติ ดังนั้นในส่วนนี้จะออกแบบให้ใช้งานคำสั่ง <path ...> ของ SVG และส่วนของเส้นบอกมิติเราจะออกแบบเป็นวัตถุเนื่องจากจะใช้งานซ้ำอีกในขั้นอื่น ๆ ได้



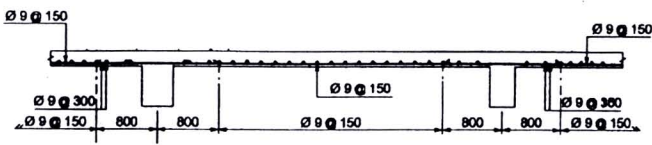
ภาพที่ 14 แบบรายละเอียดขอบนอกของแผ่นพื้น

ข. เหล็กเสริมบน ซึ่งพิจารณาแล้วจะเห็นว่าทั้งทางด้านซ้ายและด้านขวาเหมือนกัน จึง ออกแบบเป็นฟังก์ชันเดียวกัน เปลี่ยนแต่ตำแหน่ง x , y



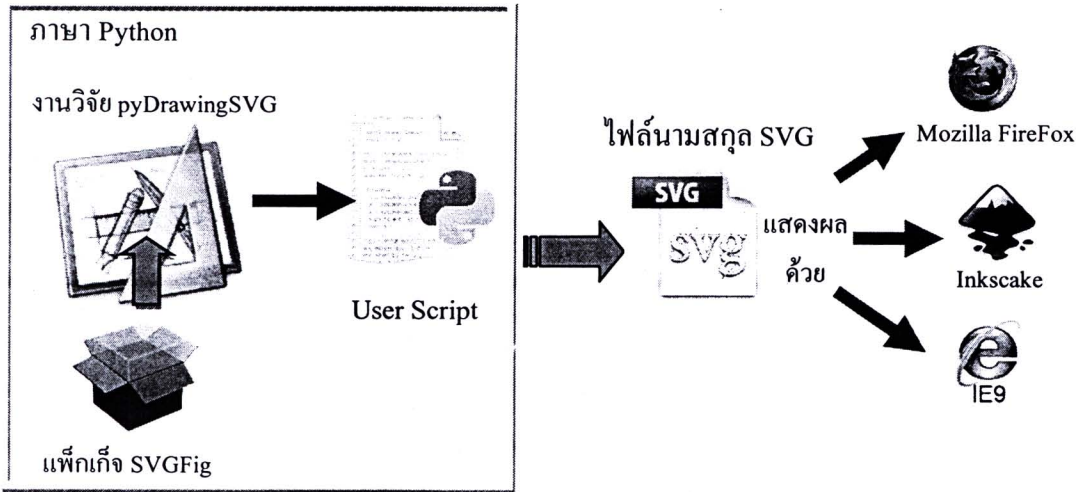
ภาพที่ 15 แบบรายละเอียดการเสริมเหล็กบนของแผ่นพื้น

ก. เขียนเหล็กเสริมล่าง จะใช้งานฟังก์ชันของเหล็กเสริม Reber ที่เราจะออกแบบไว้ ไม่ว่าจะเป็น โมดูล RebarShape หรือ โมดูล RebarSpacing



ภาพที่ 16 แบบรายละเอียดการเสริมเหล็กล่างของแผ่นพื้น

ภาพรวมของดำเนินการคือเราจะสร้างส่วนแพ็คเกจ pyDrawingSVG โดยใช้แพ็คเกจเสริม SVGFig ในการสร้างโครงสร้างภาษา SVG แล้วเมื่อต้องการใช้งานเราจะเขียนเป็นภาษาสคริปต์ของผู้ใช้งาน(User Script) เมื่อทำการสร้างเสร็จก็จะได้ไฟล์ SVG ที่นำไปดูผลได้ด้วยโปรแกรมต่าง ๆ เช่น เว็บเบราว์เซอร์ ในปัจจุบันที่กำลังทำการวิจัยจะใช้ Mozilla FireFox ในการดูผลภาพได้ ส่วน Internet Explorer ปัจจุบันอยู่ที่รุ่น 8 ยังไม่สามารถดูผลไฟล์ SVG ได้โดยตรงต้องติดตั้งส่วนเสริมของ Adobe SVG Viewer เสียก่อน แต่อย่างไรก็ตาม บริษัทไมโครซอฟท์ประกาศว่าในโปรแกรม Internet Explorer รุ่น 9 จะสามารถใช้งาน SVG ได้ (เนื่องจาก SVG เป็นมาตรฐานของ HTML5)



ภาพที่ 17 แผนภาพแสดงส่วนประกอบของแพ็คเกจ pyDrawingSVG