Carson Roscoe January 17th, 2016

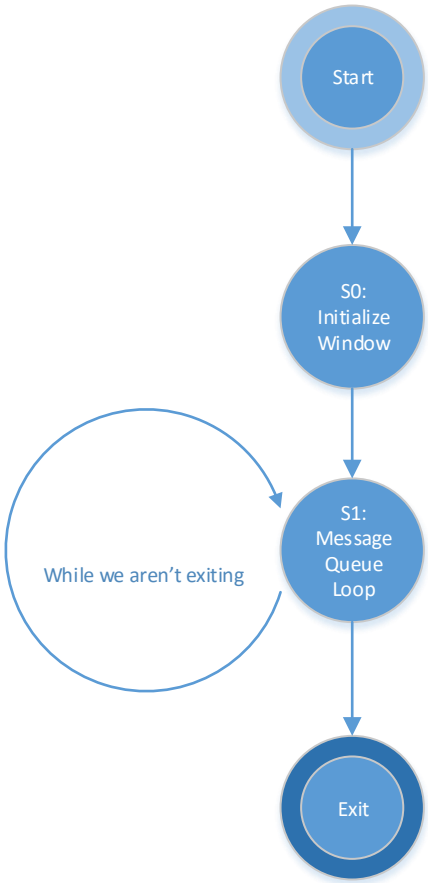
COMP4985 Design Work

Contents

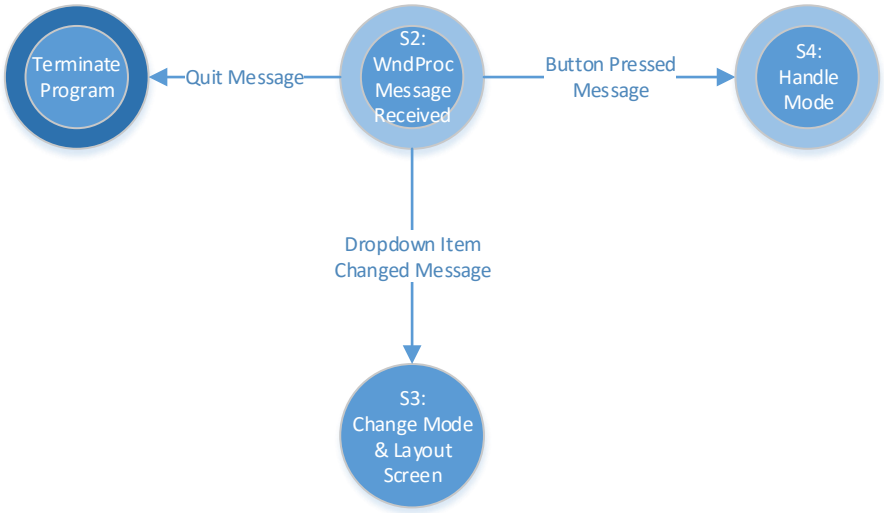
State Diagrams 2
STARTING POINT STATES (WINMAIN) 2
MESSAGE DRIVEN STATES (WNDPROC) 3
HANDLE MODE STATES 3
Pseudocode 4
STARTING POINT STATES (WINMAIN) 4
MESSAGE DRIVEN STATES (WNDPROC) 4
HANDLE MODE STATES 5

State Diagrams

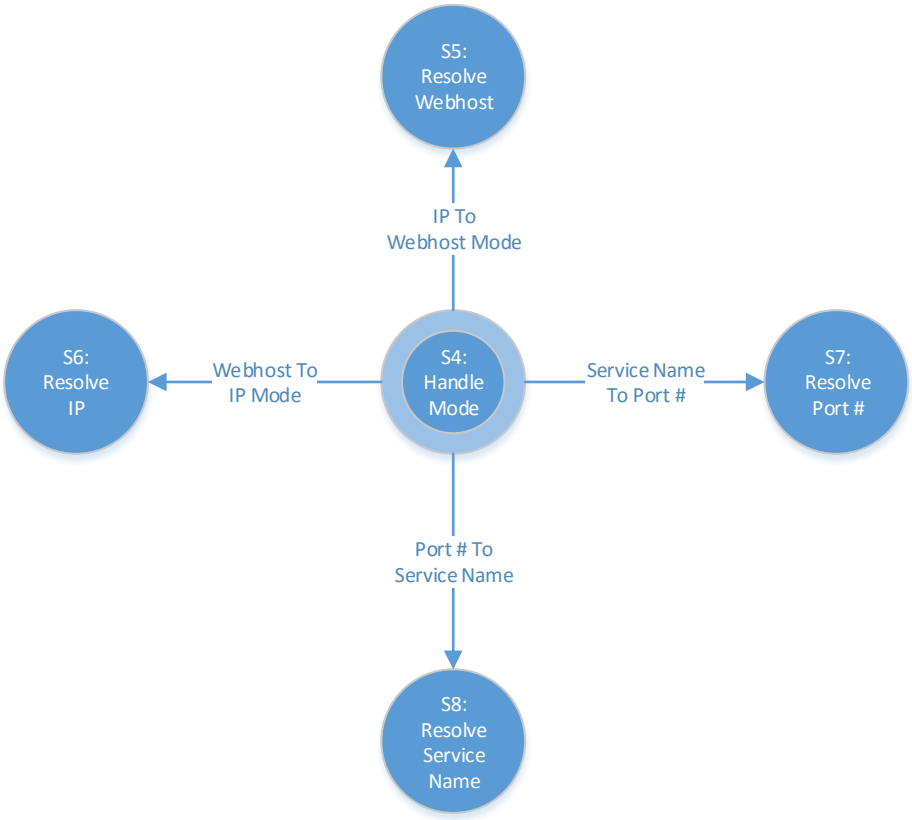
STARTING POINT STATES (WINMAIN)



MESSAGE DRIVEN STATES (WNDPROC)



HANDLE MODE STATES



Pseudocode

STARTING POINT STATES (WINMAIN)

S0: Initialize Window:

Create Window and pass in WndProc as its message handling function
Create references to all UI functions
Layout items in default manner (Default being whichever is the default mode's layout)
GOTO S1:

S1: Message Handling

While we aren't exiting the program
    Handle Win32 messaging system
Exit Program

MESSAGE DRIVEN STATES (WNDPROC)

S2: WndProc Receives Message

Switch message
    Case button pressed
        GOTO S4
    Case dropdown item changed
        GOTO S3
    Case exit program
        Exit Program

S3: Change Mode & Layout Screen

Switch new dropdown item selected
    Case Website to IP address
        Layout UI items accordingly
    Case IP address to website
        Layout UI items accordingly
    Case Service name and protocol to port number
        Layout UI items accordingly
    Case port number and protocol to service name
        Layout UI items accordingly

HANDLE MODE STATES

S4: Handle Mode

Switch case applications mode
- Case website to IP address mode
  - GOTO S5
- Case IP address to website mode
  - GOTO S6
- Case service name & protocol to port number
  - GOTO S7
- Case port number & protocol to service name
  - GOTO S8

S5: Resolve Webhost

Startup windows socket session
Allocate memory for addr_p
Store result of inet_addr command into addr_p
Store result of gethostbyaddr command into a hostent struct
If it failed, error message
Foreach address pointer in the address list provided by hostent struct
- Store aliases to a vector

MainSite equals the h_name property of the hostent struct
Close windows socket session
Update UI with MainSite & Aliases

S6: Resolve IP

Startup windows socket session
Allocate memory for addr_p
Store result of inet_addr command into addr_p
Store result of gethostbyaddr command into a hostent struct
If it failed, error message
Foreach address pointer in the address list provided by hostent struct
- Store IP to a vector
- Store aliases to a vector

MainSite equals the h_name property of the hostent struct
Close windows socket session
Update UI with MainSite, Aliases & IPs

S7: Resolve Port #

Startup windows socket session
Store result of getservbyname command into a servent struct
- If it failed, error message

Close windows socket session
portNumber equals servent struct's s_port variable
portNumber equals ntohs(portNumber)
Update UI with portNumber

S8: Resolve Service Name

Startup windows socket session
Store result of getservbyport command into a servent struct
    If it failed, error message
Close windows socket session
serviceName equals servent struct's s_name variable
Update UI with portNumber