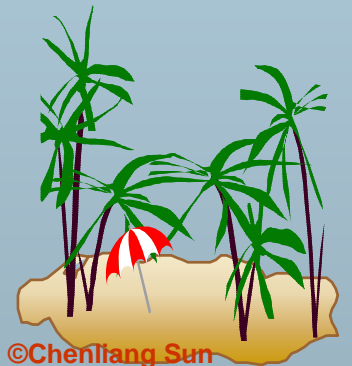




# CD Player Implementation: A simple example for Statecharts Based GUI Design

Chenliang Sun  
csun1@cs.mcgill.ca  
School of Computer Science  
McGill University

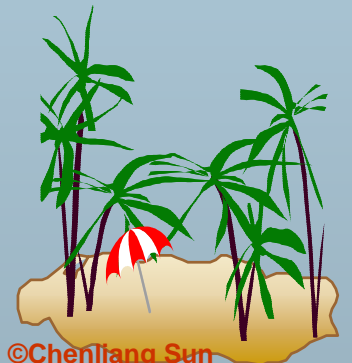
April 7, 2003

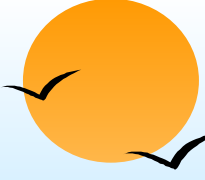




# Overview

- Why Statecharts Based GUI Design ?  
: in Theory
- The Case
- What We have Done without Considering Statecharts
- The Problems
- Coding Based on Statecharts
- Why Statecharts Based GUI Design ?  
: in Practice
- Conclusion
- References

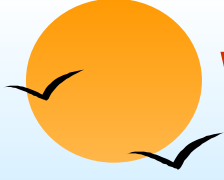




# Why Statecharts Based GUI Design? : In Theory

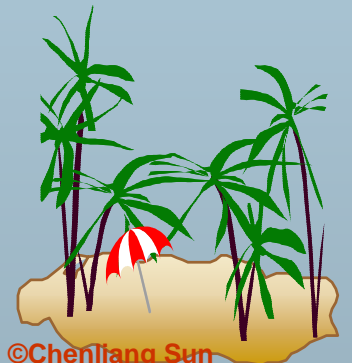
- GUIs are intrinsically far more complicated than command line interfaces because a user can have several partially completed dialog that can be suspended and resumed at any time.
- GUI must ensure that a user can only perform valid operations. e.g. Rename => name of file not blank.
- GUIs contain more bugs and are usually more difficult to test and enhance than other types of code in a system.
- .....





# Why Statecharts Based GUI Design? : In Theory

- Using Statecharts:
  - ★ much more powerful and expressive state based language
  - ★ Raise UI development from a coding task to a software design task
- Code can be written quickly and easily. Even more:  
AUTOMATICALLY CODE GENERATION !
- Easy to test using white box techniques
- Easy to enhance repeatedly over the lifetime of a system
- Can be modified without introducing unwanted side-effects



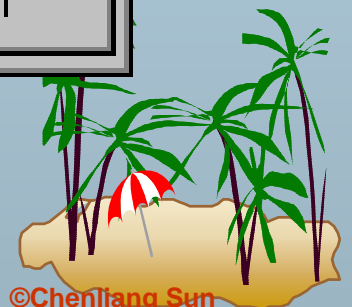


# Case Study : CD Player

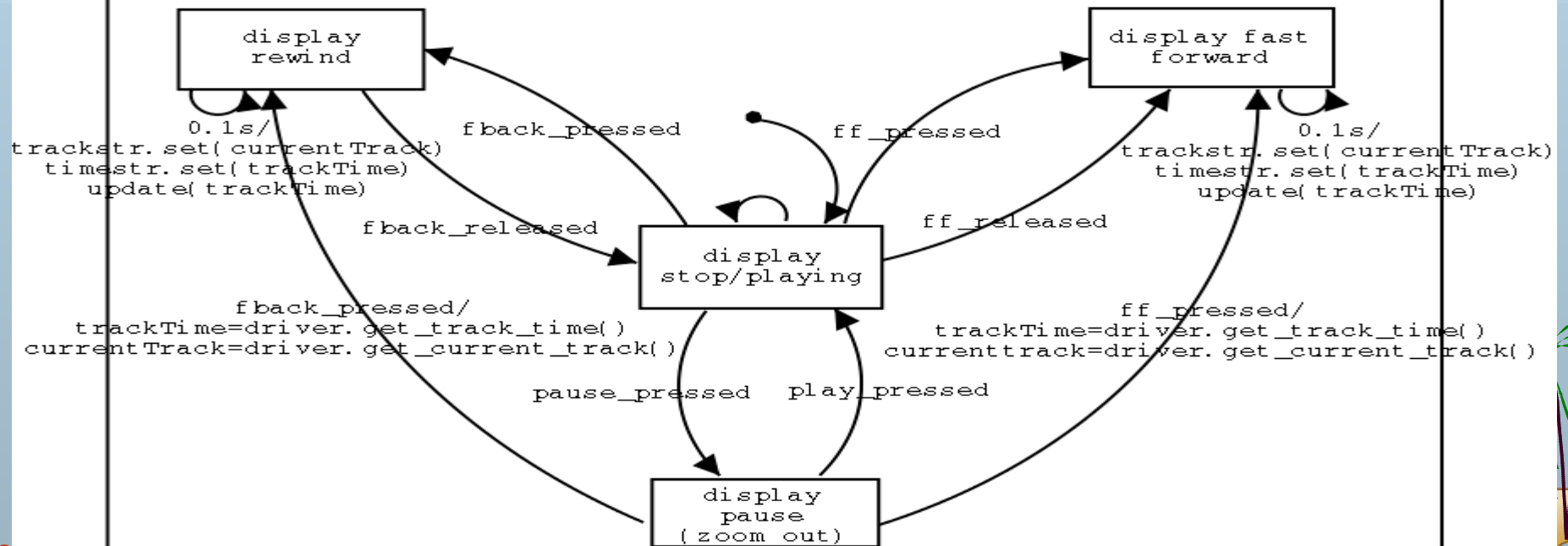
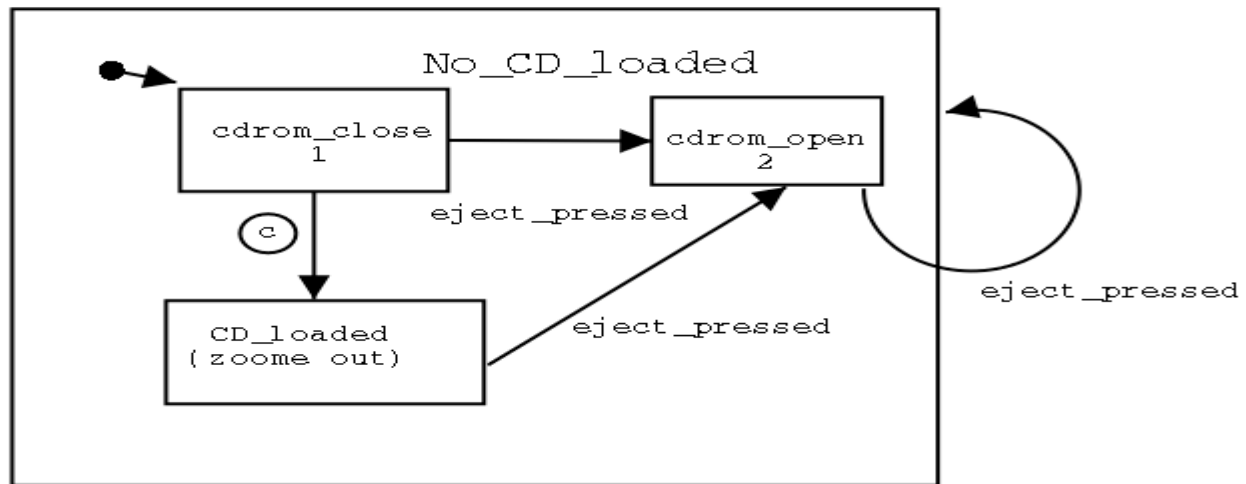
## High-level Requirements:

<http://moncs.cs.mcgill.ca/people/hv/teaching/MS/assignments/assignment3/>

Overview of the user interface screen



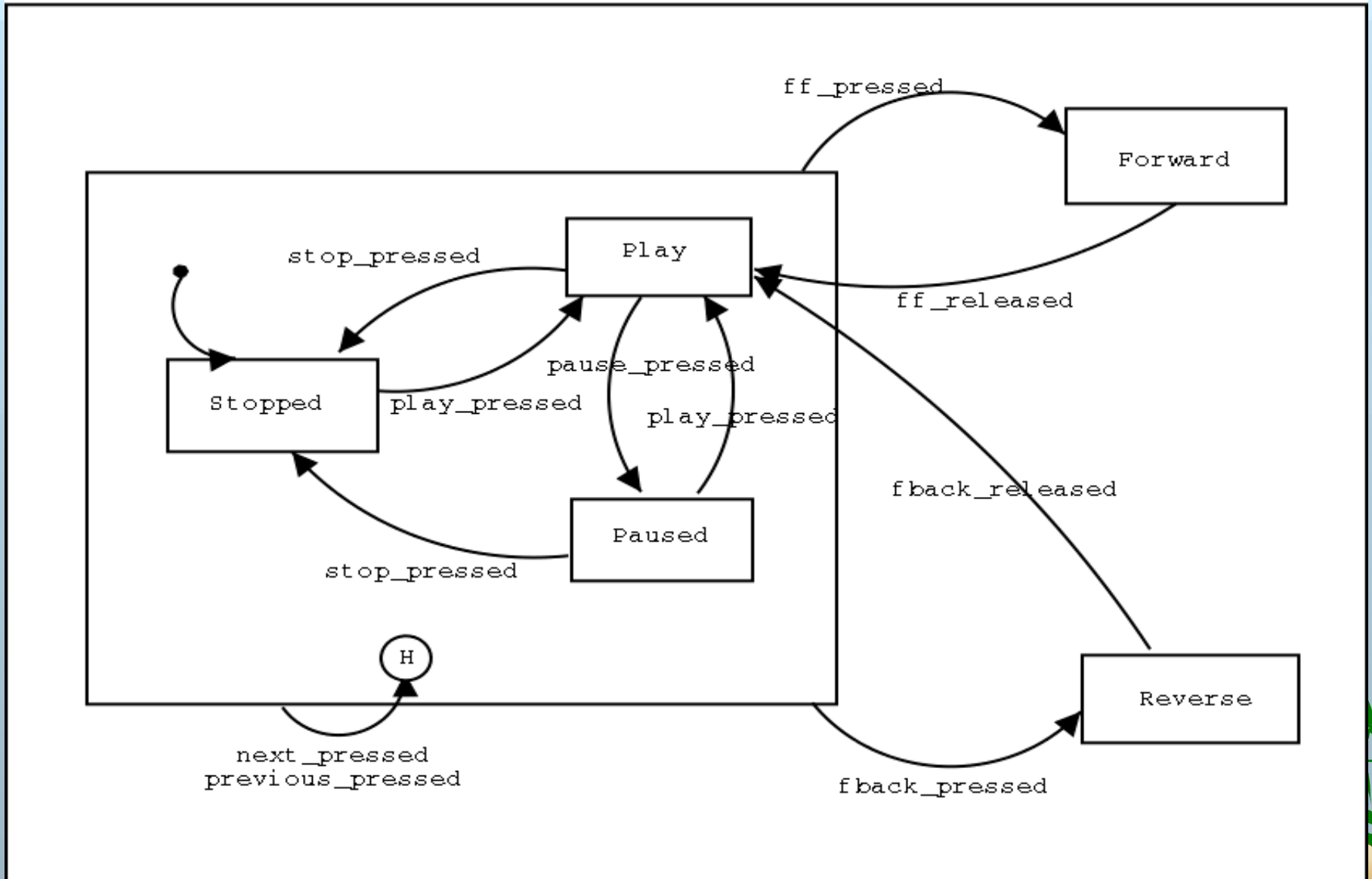
# Active





# Case Study : CD Player

CD\_loaded



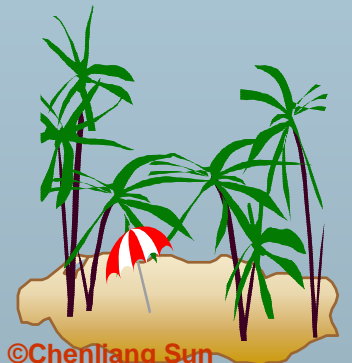


# What we have done without using statecharts

<http://moncs.cs.mcgill.ca/people/hv/teaching/MS/assignments/assignment3/>

<..\..\Python22\762\CDplayerGUI0.py>

<..\..\Python22\762\CDplayer0.py>

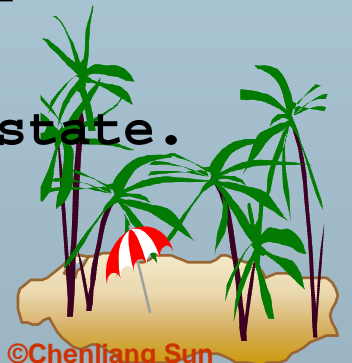






# The Problems

- It took one member about 3 days.
- It took me one day to debug.
- Still have problems:
  - In start, when click eject button, the CD player shows “No track” . But when the drawer is closed and with CD in, still shows that info.
  - After pressed previous or next button, it should go to history state, but the CD player goes to playing state.
  - Press fast forwards, text doesn't change, but when release, it fast forwards. It seems unwilling to stop so you have to press stop button.
  - Press >> then >>|, should go to history state. But the CD player still fast forwards.
  - may contains more bug ☹.



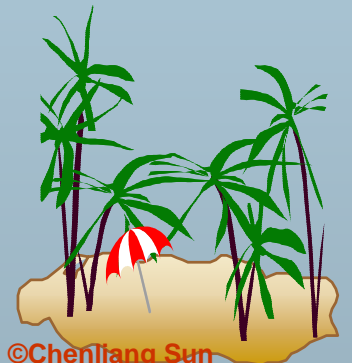


# Coding a statechart

Coding a statechart is a simple process.

There are 4 main tasks which should be carried out in the following order:

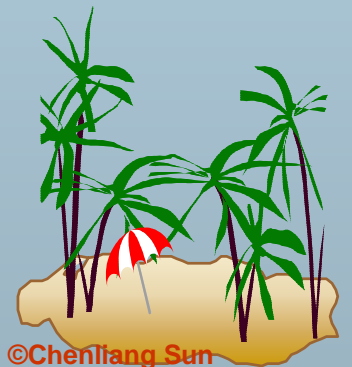
- Create the user interface objects (ideally this should be done by an interaction designer)
- Create state variables
- Create the state procedures
- Implement the state transitions defined in the event-action tables of the statecharts





# Coding based on statecharts

- New codes:
- [..\..\..\Python22\762\CDplayer.py](#)
- [..\..\..\Python22\762\CDplayerGUI.py](#)





# Automatic Code Generation

## ■ From: executable specification

executable UML

Statecharts

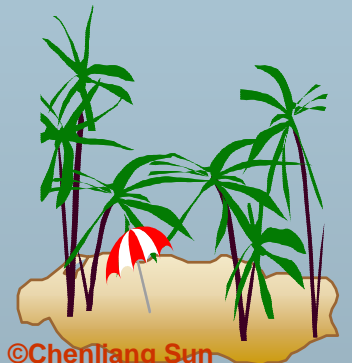
Stateflow

colored Petri Net

.....

Statechart is one of the most powerful formalisms. There are some products which generate code from statecharts

- + Allow developers to work out higher levels of abstraction on design and specification
- + Human error in the coding phase can be eliminated





# Why Statecharts based design: In practice

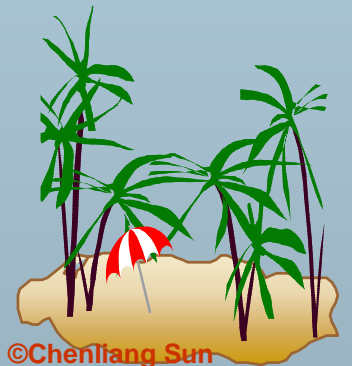
- Easy to implement
- Easy to test
- Easy to maintain





# Conclusion

- Statecharts provide much more powerful and expressive state based language, which is much more concise and accurate than natural language.
- Statecharts based design raise UI (more precisely, software) development from a coding task to a software design task.
- Code from statecharts can be written quickly and easily. Even allows automatic code generation.
- easy to test using white box techniques.
- ..... •





# References

- Ian Horrocks, *Constructing the User Interface with Statecharts*, Addison-Wesley, 1998
- Hans Vangheluwe, *Modeling and Simulation course lecture notes*, School of Computer Science, McGill University, 2002

