

---

# Testing Web Applications

---

Shilpi Narula

M.Sc. Mathematics

Dayalbagh Educational Institute

Dayalbagh, Agra

---

# Web Applications: Characteristics

- Web applications do not just give us new types of applications, but provide an entirely new way to deploy software solutions to end users.
- Web applications employ a number of new languages, technologies, and programming models.

---

# Cont..

- Modern Web applications are sophisticated, interactive programs with complex GUIs and numerous back-end software components that are integrated in novel and interesting ways.
- Web applications are much more complicated than simple HTML Web pages, and consist of more than just the front-end graphical user interfaces that users see.

---

# Problems in Testing

- In traditional client-server systems, the respective roles of the clients and servers and their interactions are predefined and static.
- In web applications, however, client side programs and contents may be generated dynamically.
- Compatibility and interoperability are urgent and cause problems that are more serious than with traditional programs.

---

# Cont..

- Web applications also have much faster maintenance requirements than most traditional software.
- Web technologies evolve more rapidly than traditional software technologies.
- Web applications also have features that are not present in client-server and distributed systems. These include session control, cookies, and the stateless aspect of HTTP.

---

# Cont..

- There are a wide number of users distributed all over the world & accessing the application services in a concurrent and global way.
- There is a possibility to generate software components at run time.
- The components may be executed on different server platforms .
- Different browsers may be used.

---

Effective testing of web based applications has to rely on [5]:

- Test Models
- Test Strategies
- Testing Levels
- Testing Processes

---

# Modeling Web-based Applications.

Models represent the components of the application under test and their interconnections

- UML Model (F. Ricca and P. Tonella [1, 2, 3])
- Finite State Machine Model.(Anneliese A. Andrews, Je Offutt ,Roger T. Alexander [4])

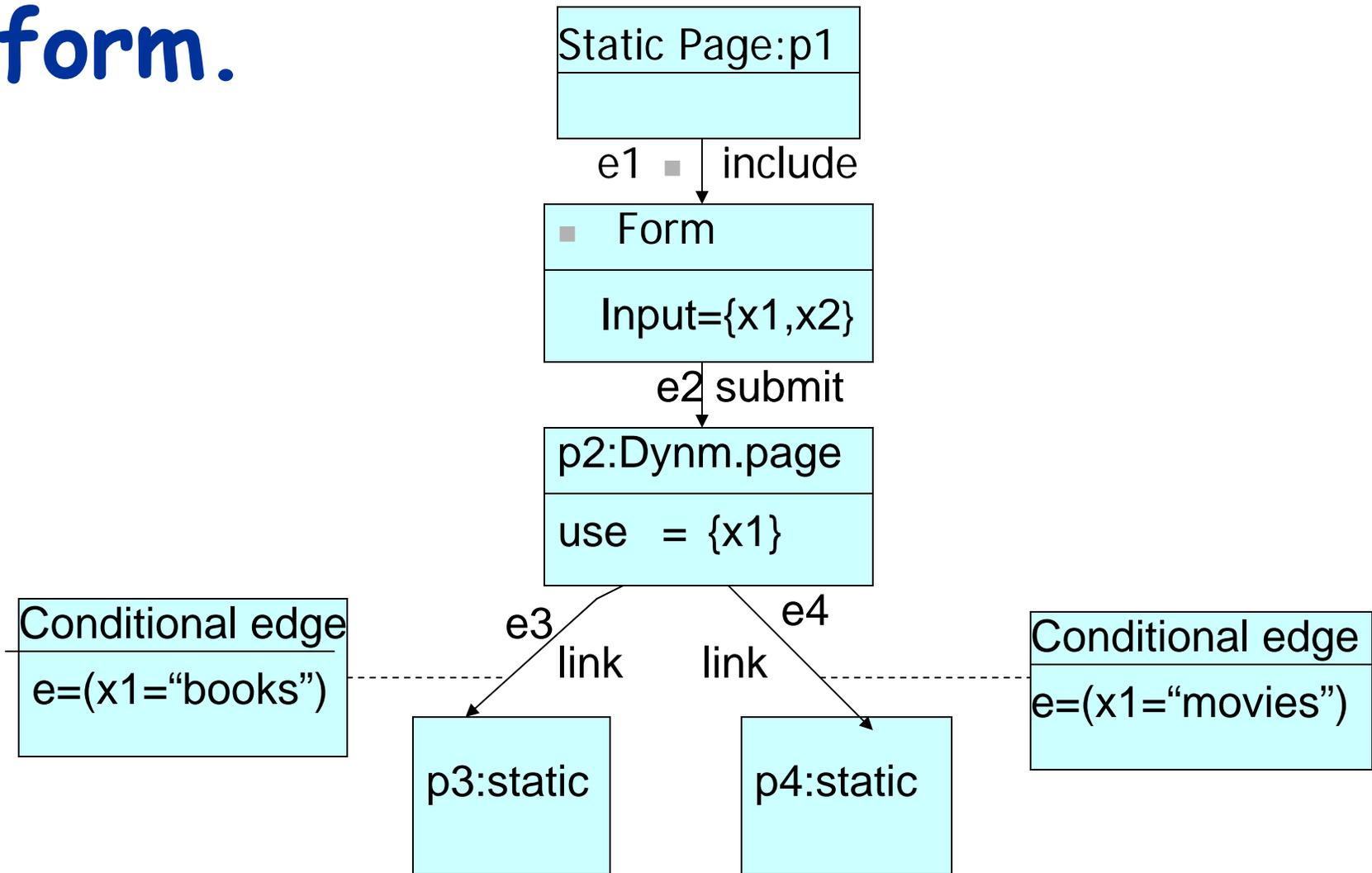
---

# Modeling Web Applications using UML

---



# Example of model with a form.



---

# Testing

There are two types of testing techniques:

- ❑ Static Verification.
- ❑ Dynamic Validation.

---

# Static Verification.

- These techniques do not require execution.
- They work on artifacts such as design documents and HTML source.
- Static analyzers can be employed to scan the HTML pages in a Web site & detect possible faults.

---

# Cont...

- The model of the site can be analyzed to determine the presence of **unreachable pages**.
- **Ghost pages** are associated with pending links, which reference of a non existing frame.
- The analyses of a **reaching frames** is a specialization of the flow analysis framework, which computes the set of frames in which each page can appear .
- The outcome of the reaching frames is useful to understand the assignment of pages to frames.

---

## Cont...

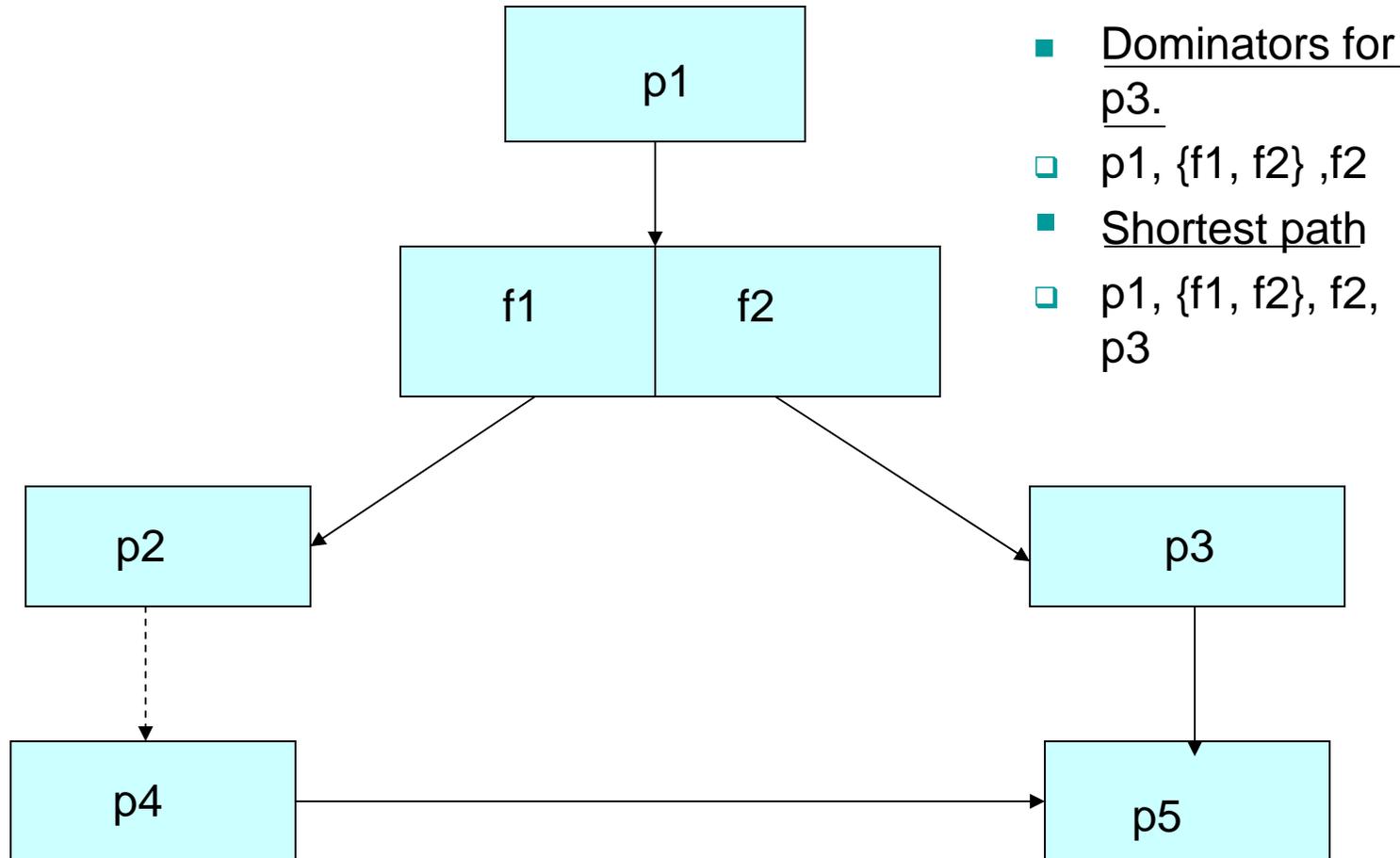
- Flow analyses can be employed in a more traditional fashion to determine the **data dependences**.
- Data dependences may reveal the presence of undesirable possibilities, such as use of a variable not yet defined.
- These are also important for dynamic validation.

---

# Cont...

- While traversing the web pages it is impossible to reach a given document without traversing a set of other pages, called its **dominators**.
- Information about the **shortest path** is an indicator of potential troubles for the user searching a given document.

# Example of a model with a frame



---

# Dynamic validation

- The internal structure of a Web application is accessed to measure the coverage that a given test suite reaches, with respect to a given test criterion.
- A test case for a Web application is a sequence of pages to be visited plus the input values to be provided to pages containing them.
- Execution consists of requesting the Web server for the URLs in the sequence and storing the output pages

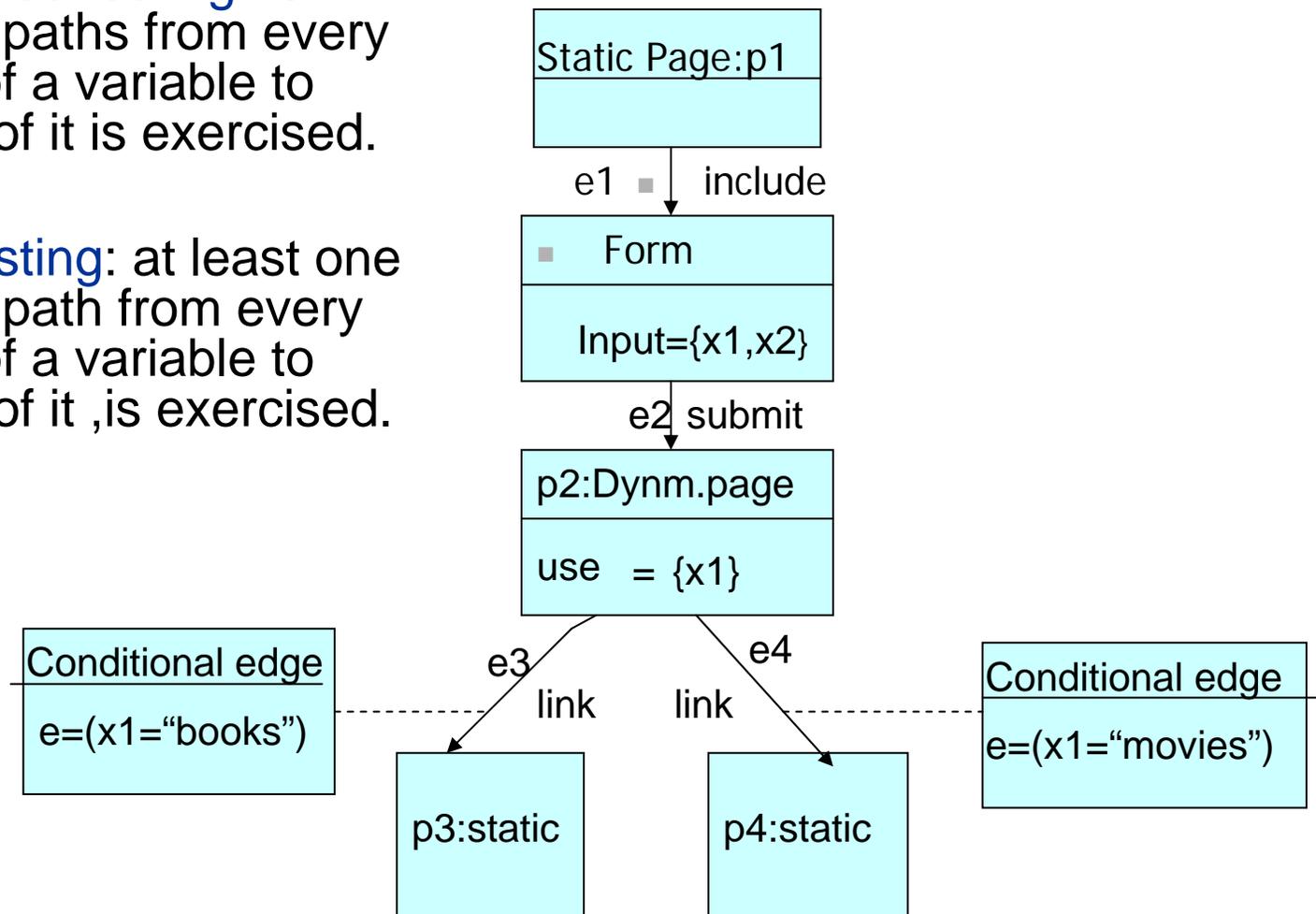
---

# Testing Criteria

- **Page testing**: every page is visited at least once in some test case.
- **Hyperlink testing**: every hyperlink from every page in the site is traversed at least once.
- **All-paths testing**: every path in the site is traversed in some test case at least once.

# Cont...

- **Definition-use testing** : all navigation paths from every definition of a variable to every use of it is exercised.
- **All-uses testing**: at least one navigation path from every definition of a variable to every use of it ,is exercised.



---

# Cont...

- These criteria are often impractical, since there are infinite paths in a site.
- They can be satisfied if additional constraints are imposed ,for example k-limiting and path independence.
- In the k-limiting case , loops are traversed at least k times.
- In the second case only independent paths are considered.
- When designing & executing test cases for a web application not all pages are equally of interest, Static pages can be ignored.

---

# Applicability

- Two tools **ReWeb** and **TestWeb** have been developed by Ricca and Tonella [2, 3, 4 ] to support analysis and testing of Web applications.
  - **ReWeb** downloads and analyzes the pages of a Web application with the purpose of building a UML model of it.
  - **TestWeb** generates and executes a set of test cases for a Web application whose model was computed by **ReWeb**.
- The Analysis and Testing approach has been applied to several real world web applications, e.g., Amazon

---

# Modeling Web Applications using FSMs

---

---

# Definitions

- *Cluster* : Collections of software modules and Web pages that implement some logical function.
  - At the highest level, clusters may be abstractions that implement functions that can be identified by users.
  - At the bottom level, clusters may be individual Web pages and software modules that represent single major functions.
- Web pages are modeled as multiple *Logical Web Pages (LWP)*.
  - An LWP is either an entire physical Web page or the portion of a Web page that accepts data from the user through an HTML form and then sends the data to a specific software module.

---

# FSMs

- Partitioning results in a hierarchical collection of finite state machines.
  - At the lowest level, logical Web pages are represented by nodes in FSMs that model behavior of software modules and Web pages.
  - Edges represent transitions among logical Web pages and software modules.
  - At the higher levels, each cluster forms a node in a FSM that models behavior of clusters.
  - Edges in both levels of FSMs are annotated with inputs and constraints that may be associated with the transition.

---

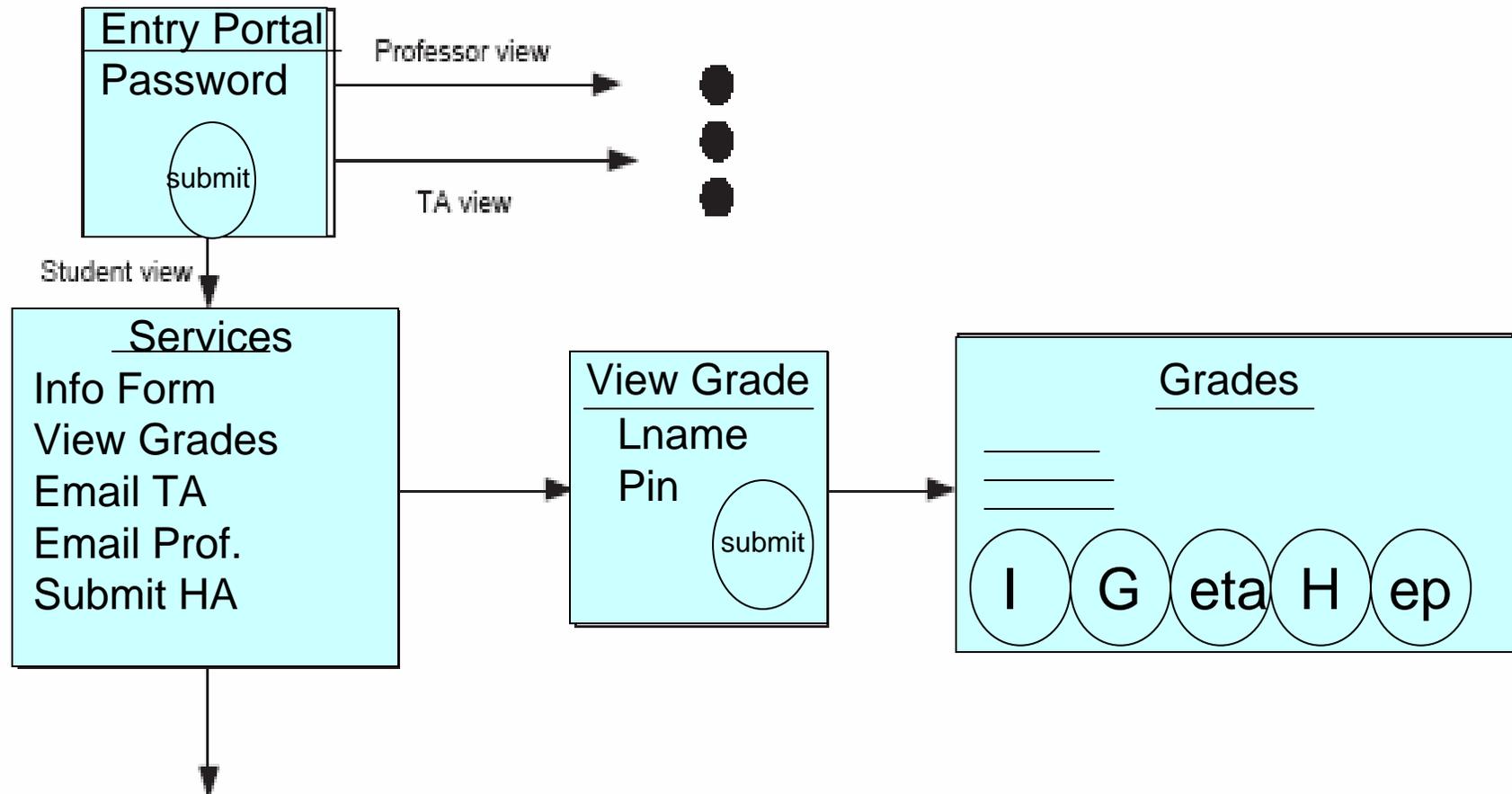
# Testing Approach

- *Phase 1*: Build a model of the Web application.
  - Partition the Web application into clusters,
  - Define logical Web Pages,
  - Build FSMs for each cluster,
  - Build an FSM to represent the entire Web application.
- *Phase 2*: Generate test from the model defined in Phase 1
  - Generate tests as sequences of transitions,
  - Identify values for inputs.

# An Example Web Application: CSIS

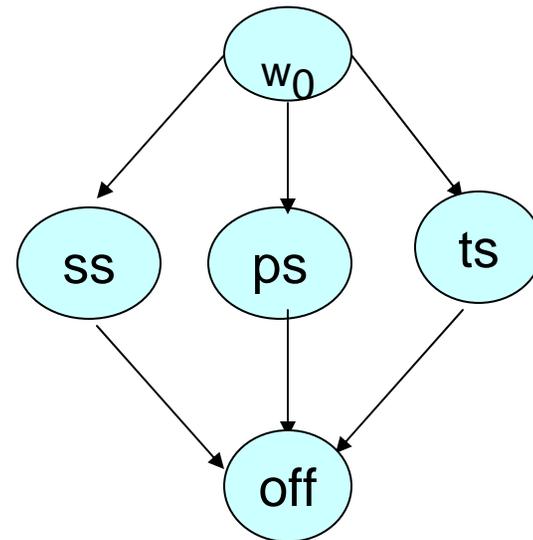
Student	Professor	Teaching assistant
Information System	View Student Information	Post Grades
View grades	Post Grades	Email Class
Email TA	Email Grades	Email Student
Email Professor	Email Student	Email Professor
Submit HA	Email TA	View Course Information
Course Information	Post Course Information.	

# Course Student Information System (CSIS)



# Top level FSM

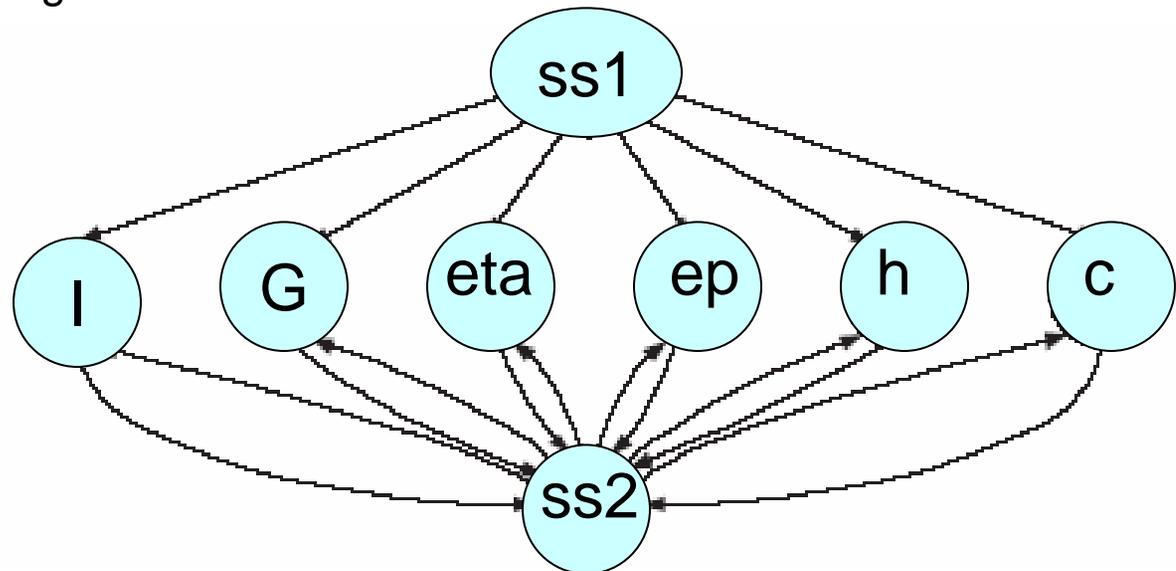
- SS: Student Services
- PS: professor Services
- TA: Teaching Assistant Services
- $W_0$ : Login Screen
- off: Logoff



# Student Services

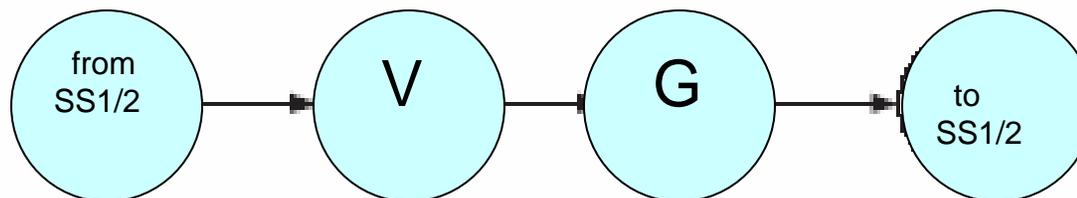
## ■ Node Cluster Explanation

- ❑ SS1: Select Service 1 - Menu of named services
- ❑ SS2: Select Service 2 - List of navigation buttons for services at the bottom of student service pages
- ❑ I: Info Form - Physical Info Form page without service navigation buttons at bottom
- ❑ G: Grades - View Grades and Grades pages
- ❑ ETA: Email TA
- ❑ EP: Email Prof
- ❑ H: Submit homework
- ❑ C: View course info Pages



# View Grades

From SS1/2	From Select Service 1/2	Connection from either service selection facility
to SS2	To Select Service 2	Connection to list of navigation buttons for services at the bottom of student service pages
V	View Grades	Authentication for viewing grades
G	Grades	Grade display



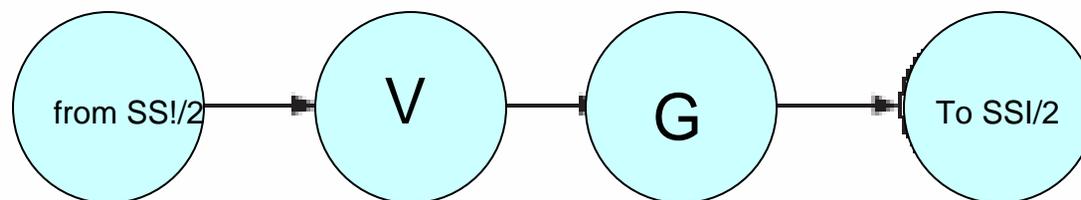
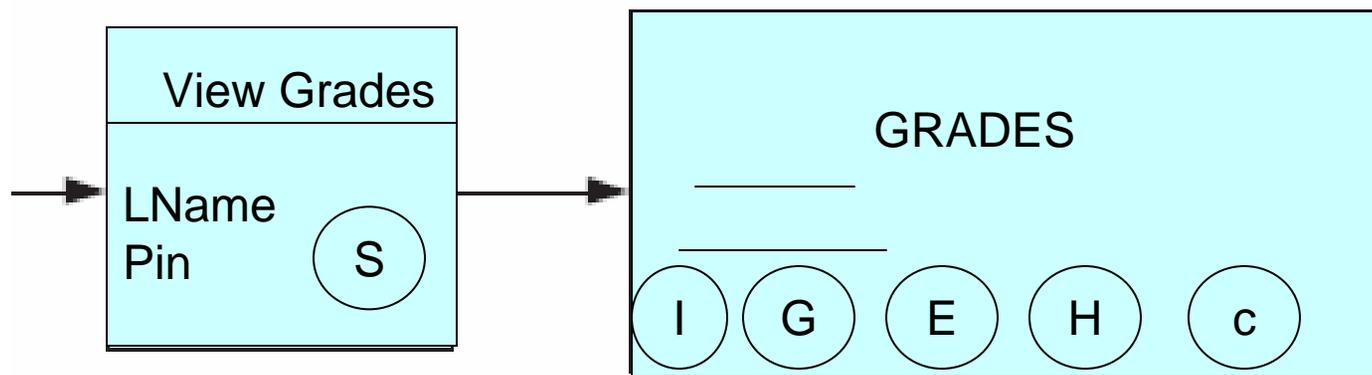
## Input Types

<b>Text</b>	<b>Non-text</b>
digit line email phone URL multi-line file	links buttons radio button drop-down list check boxes

## Constraints on Inputs

<b>Input Choice</b>	<b>Order</b>
Required ( <b>R</b> ) Required Value ( <b>R</b> (parm=value)) Optional ( <b>O</b> ) Single Choice ( <b>C1</b> ) Multiple Choice ( <b>Cn</b> )	Sequence ( <b>S</b> ) Any ( <b>A</b> )

Transition	Constraints	Alphabet encoding of Constraint
(SS1, V)	none	e
(V, G)	R (Lname, PIN, Submit) S (A (Lname, PIN), Submit)	f
(G, SS2)	none	e



---

# Generating Test Sequences

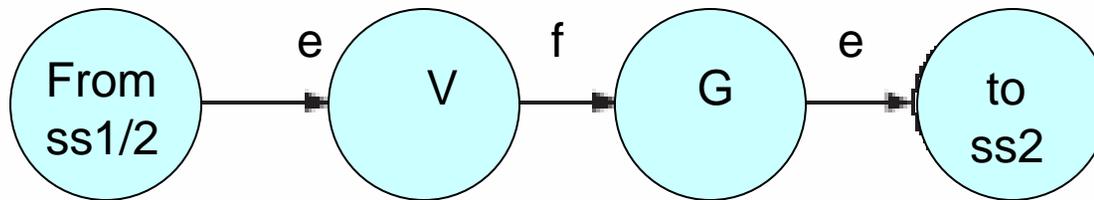
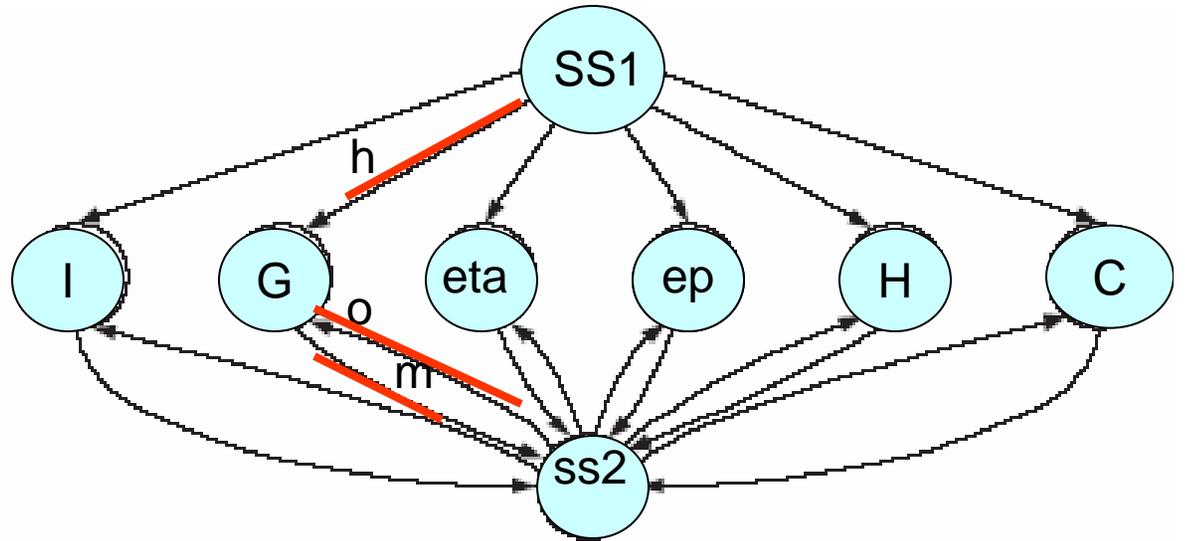
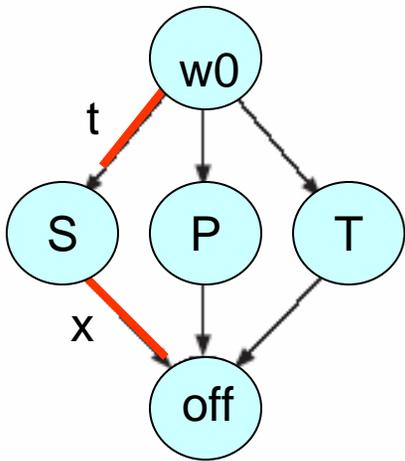
- A test sequence is a sequence of transitions in an FSM and the associated constraints.
- Assume FSM  $f$  has test sequences  $Sf_1, Sf_2, \dots Sf_N$ .
- Furthermore, suppose FSM  $f$  appears in an aggregate FSM as node  $f$ , and there is an edge in the FSM from node  $f$  to node  $g$  with annotation  $A$ .
- Generating partial aggregate test sequences
  - If test sequence  $j$  from FSM  $f$  is  $Sf_j$ , and test sequence  $k$  from FSM  $g$  is  $Sg_j$ , then a partial aggregate test sequence is  
 $Sf_j; A; Sg_k$ .

---

# Generating Test Sequences

- Full aggregate test sequences are created by combining partial aggregate sequences.
  - Assume the aggregate FSM has a test sequence of nodes  $f_1, f_2, \dots, f_n$ , with annotations  $A_{12}, A_{23}, \dots, A_{n-1n}$  and the FSMs from the nodes in the aggregate FSM test sequence have test sequences  $S_{f_1;j}, S_{f_2;j}, \dots, S_{f_n;j}$ . Then the aggregate test sequence is

$S_{f_1;j}; A_{23}; S_{f_2;j}; A_{23}; \dots; A_{n-1n}; S_{f_n;j}$ .



t; h; e; f; e; m; o; e; f; e; m; x

<b>Transition Symbol</b>	<b>Constraint</b>
h	R(G)
e	none
f	R(Lname, PIN, Submit) S(A(Lname, PIN), Submit)
e	none
m	continue-use (Lname, PIN)
o	R(G)
e	none
f	R(Lname, PIN, Submit) S(A(Lname, PIN), Submit)
e	none
m	continue-use (Lname, PIN)

---

# Problems

- The “oracle” problem, that is, of deciding if the results are correct. This is a problem because
  - of low observability
  - tracking outputs is difficult
- Dealing with unanticipated user controlled transitions
  - This includes a user going directly to an internal Web page with a bookmark or URL rewriting and unanticipated back and forward transitions

---

# Conclusions

- Web Application Testing is important
- Two different approaches to modelling and testing have been discussed
  - UML based
  - FSM based

---

# References

1. F. Ricca and P. Tonella, 'Web Site Analysis: Structure and Evolution', Proc. of *ICSM'2000, International Conference on Software Maintenance*, pp. 76-86, San Jose, California, USA, October 11-14, 2000.
2. F. Ricca and P. Tonella, 'Analysis and Testing of Web Applications', Proc. of *ICSE'2001, International Conference on Software Engineering*, pp. 25-34, Toronto, Ontario, Canada, May 12-19, 2001.
3. F. Ricca, P. Tonella. 'Detecting Anomaly and Failure in Web Applications', *IEEE MultiMedia magazine* , Vol. 13, n. 2, pp. 44-51. April-June 2006.
4. A. Andrews, J. Offutt and R. Alexander, 'Testing Web Applications by Modeling with FSMs', *Software Systems and Modeling*, Vol.4, No.3, pp.326-345, July 2005.
5. G. A. Di Lucca, 'Testing Web Applications: The State of Art and the Future Trends', *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, Vol. 2, pp.25, July 2005.

---

**THANK YOU**

---