ICWE - 2020

Automatic Model Completion for Web Applications

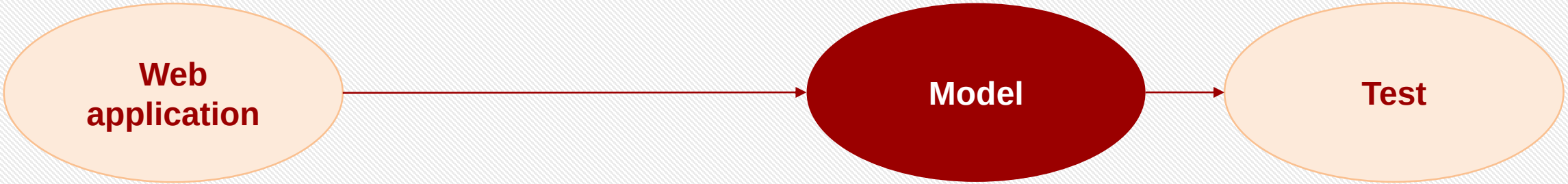
Ruilian Zhao, Chen Chen, Weiwei Wang*, Junxia Guo

Beijing University of Chemical Technology

PART ONE

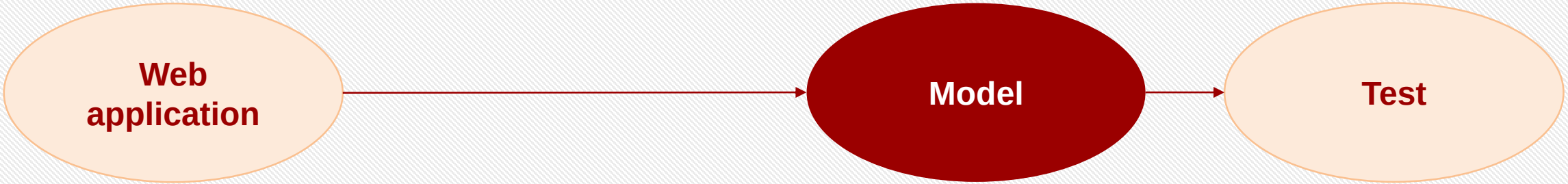
Introduction

Model construction



- Model-based testing is one of the most effective methods for testing web applications, where the integrity of models determines the effectiveness and efficiency of testing.

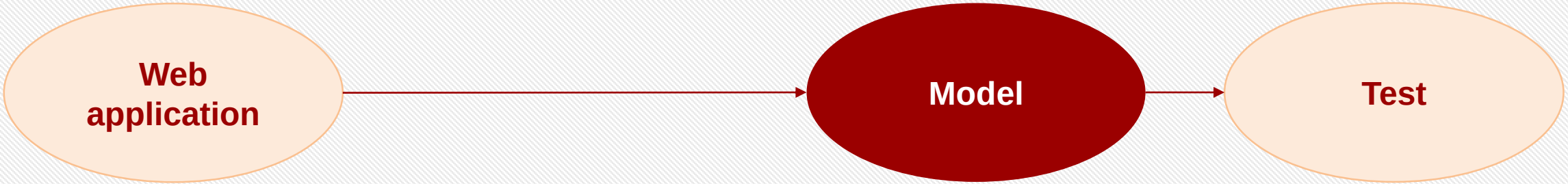
Model construction



Static analysis

Dynamic analysis

Model construction

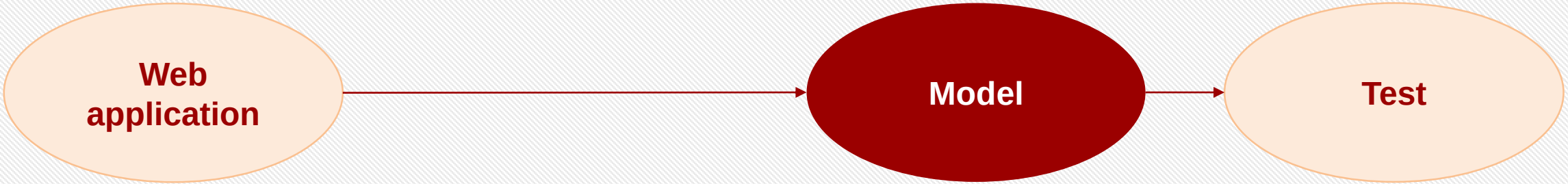


Event driven features

Dynamic features

→ Static analysis → Incomplete model

Model construction



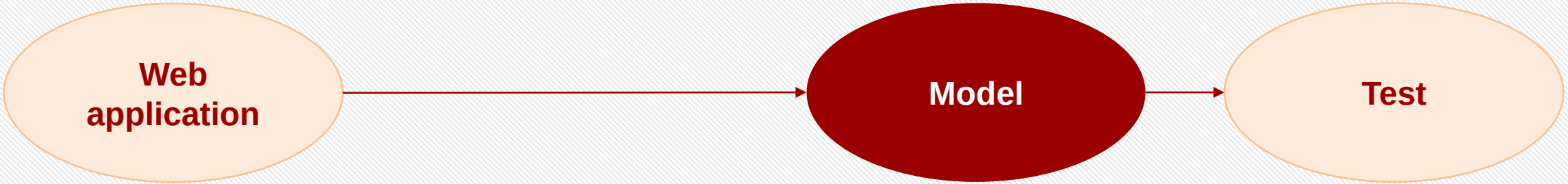
Event driven features

Dynamic features

Static analysis → Incomplete model

Dynamic analysis

Model construction



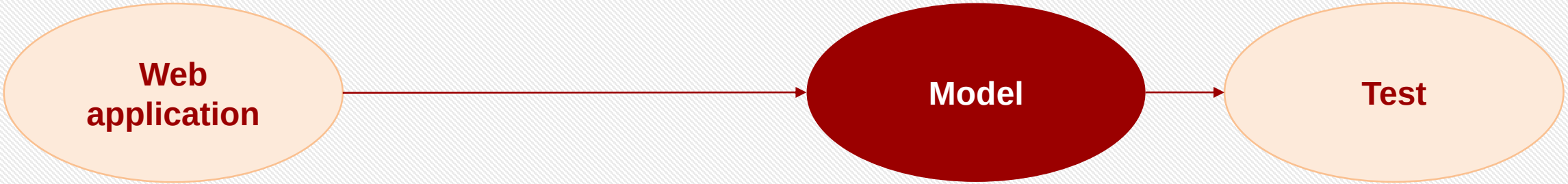
Event driven features

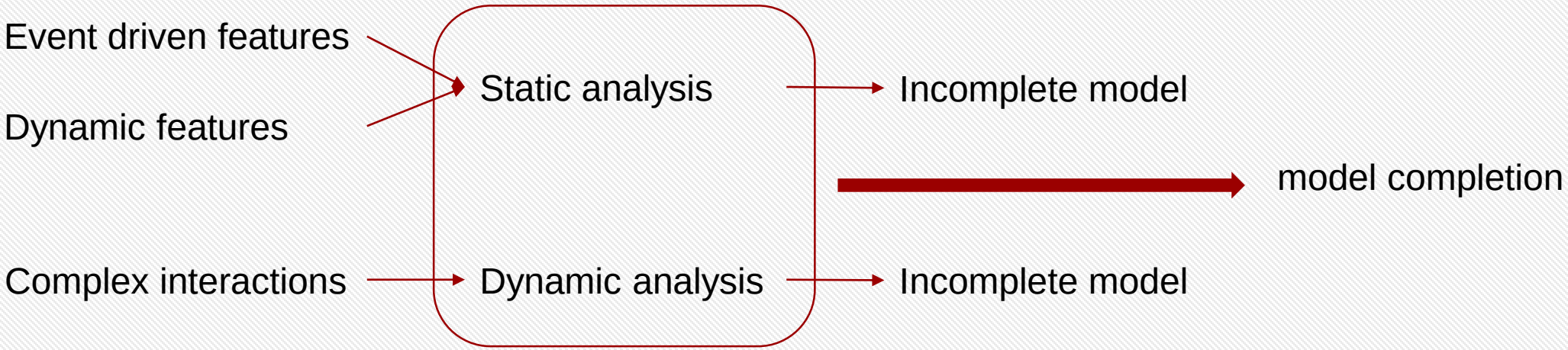
Dynamic features

→ Static analysis → Incomplete model

Complex interactions → Dynamic analysis → Incomplete model

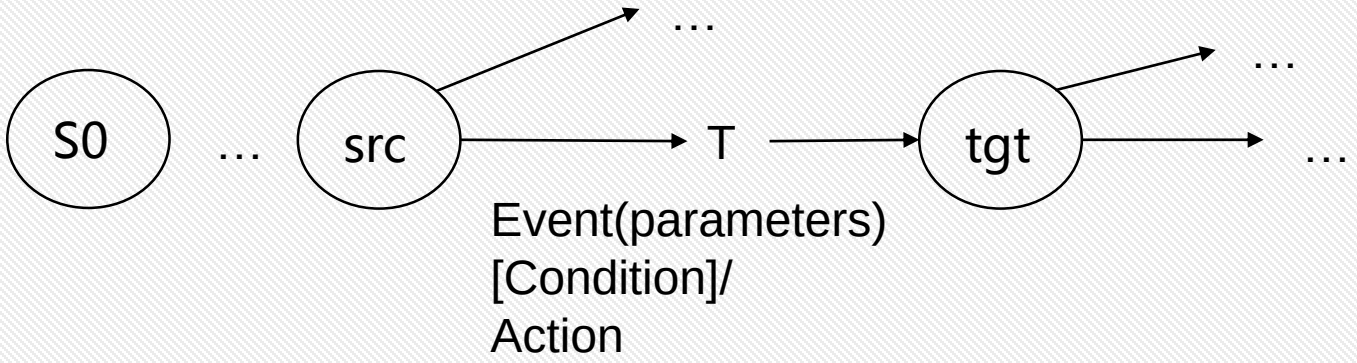
Model construction





EFSM model

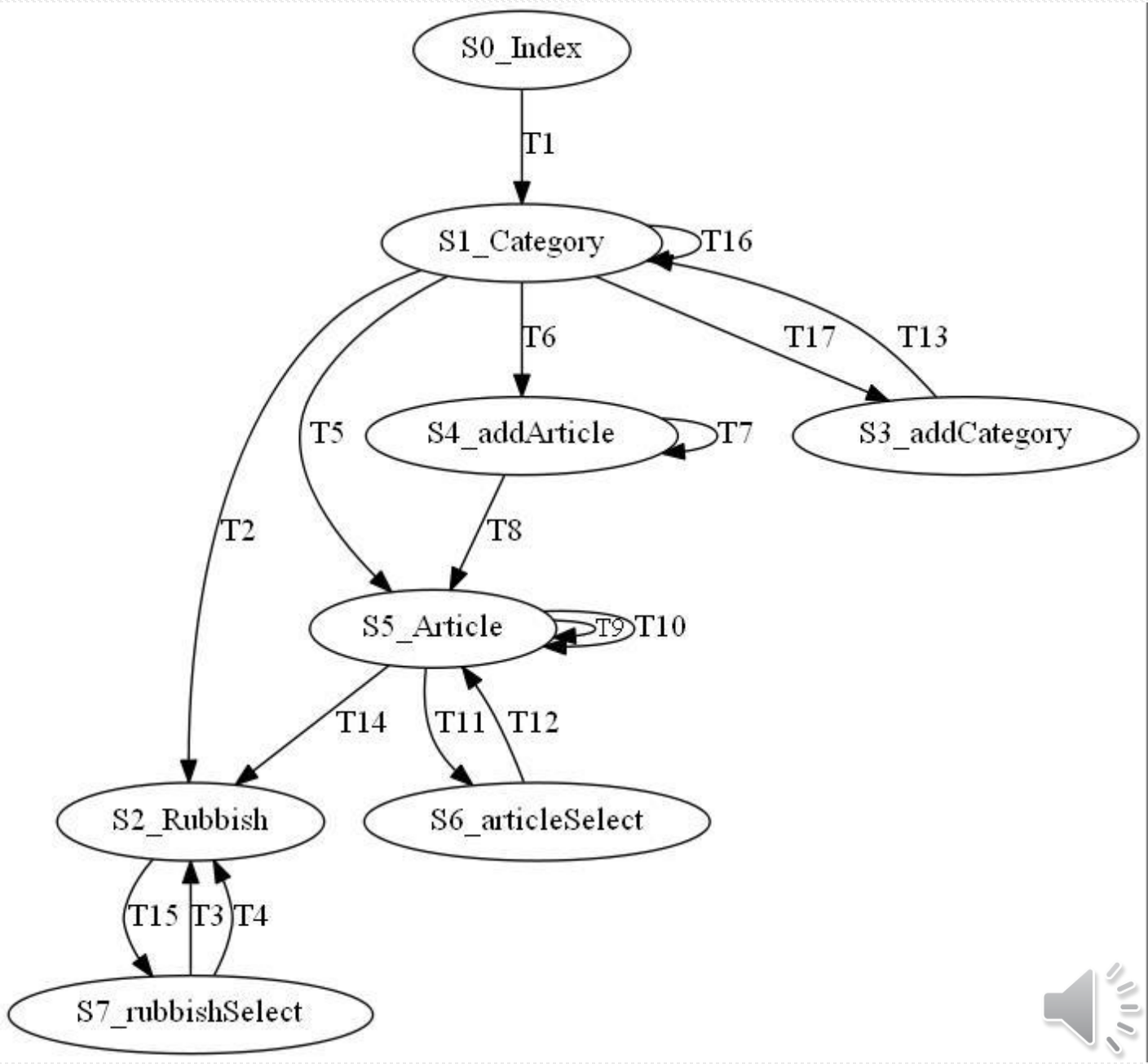
- Extended Finite State Machine (EFSM) is a widely used model that consists of states and transitions, where the states represent web pages and the transitions represent the trigger-events, trigger-conditions and follow-up operations.



Case Study

- PhpaaCMS

  ——Article management module



Case Study

- The details of transitions in the EFSM model

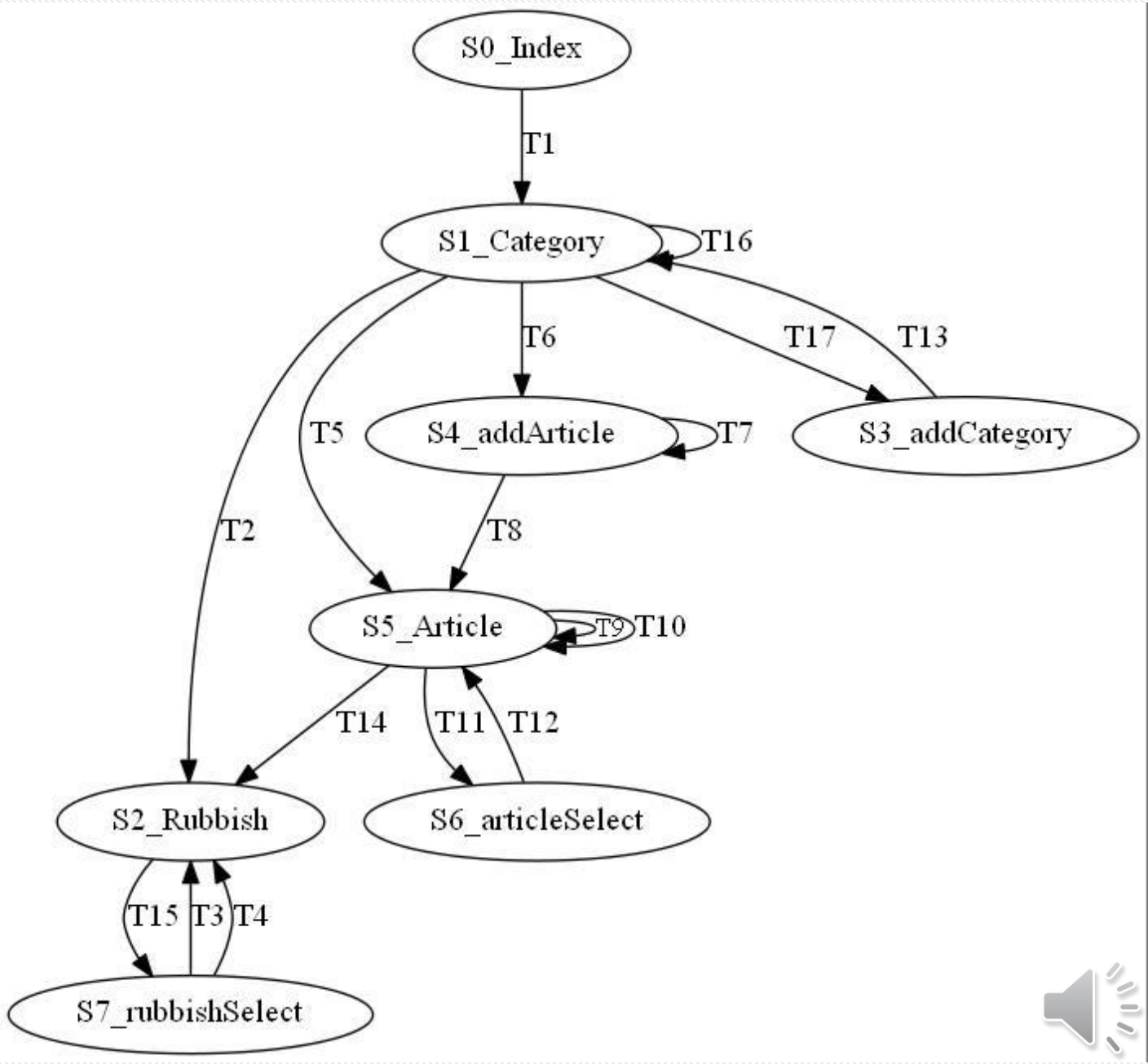
Trans	src	tgt	event	condition	action
T1	S0	S1	click,link=Admin	-	-
T5	S1	S5	click,Xpath://input[@value='Add article']	-	-
T10	S5	S5	click,Xpath:(//img[@alt='delete'])[2]	a=='delete'&&id	-
T11	S5	S6	click,name=checkbox (id)	-	getCheckedIds('checkbox')
T12	S6	S5	click,id=Xpath:(//img[@alt='deleteAll'])[2]	a=='deleteAll'&&id	-
T13	S3	S1	event:click,name=button (pid,name,seq)	-	-
T17	S1	S3	click,Xpath://input[@value='Add category']	-	-

Case Study

- PhpaaCMS

  ——Article management module

batch transfer



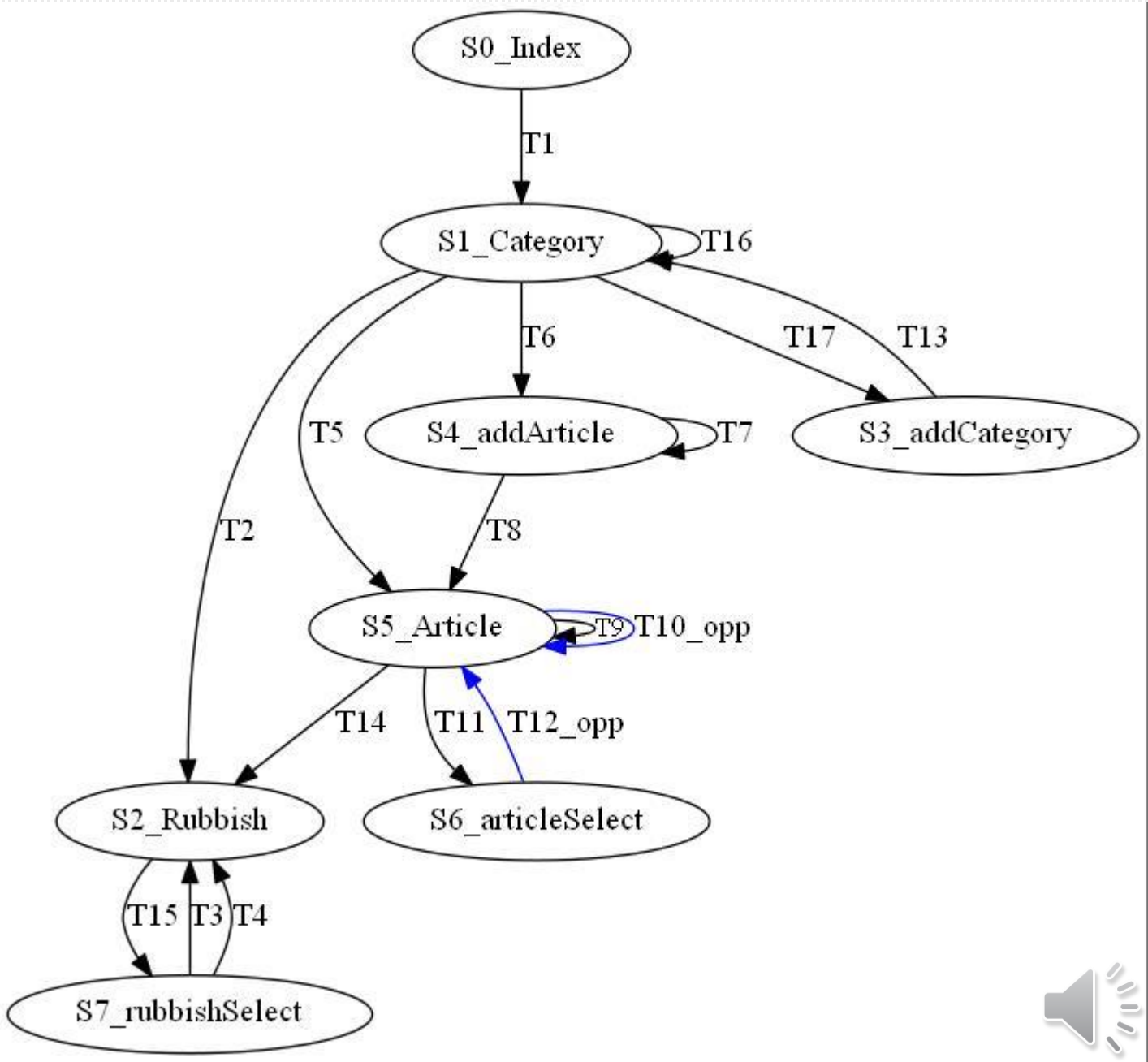
Case Study

- JavaScript code of the event handler doAction

```
1  <script type="text/javascript">
2  function doAction(a,id){
3  if(a=='deleteAll'&&id){//opposite condition
4          if(confirm('Delete all?')){ $.ajax({
5          data:'act=deleteAll&id='+getCheckedIds('checkbox'),
6                  success:function(data) {.....}//T10     });}}
7  if(a=='delete'&&id){//opposite condition
8          if(confirm('Delete?')){ $.ajax({
9                  data:'act=delete&id='+id,
10                 success:function(data) {.....}//T12 });}}
11 if(a=='moveAll'&&id){//uncovered condition
12         if(confirm('All transferred?')){$.ajax({
13             data:'act=moveAll&id='+getCheckedIds('checkbox'),
14             success:function(data) {.....}//T14 });}}}
15 </script>
```

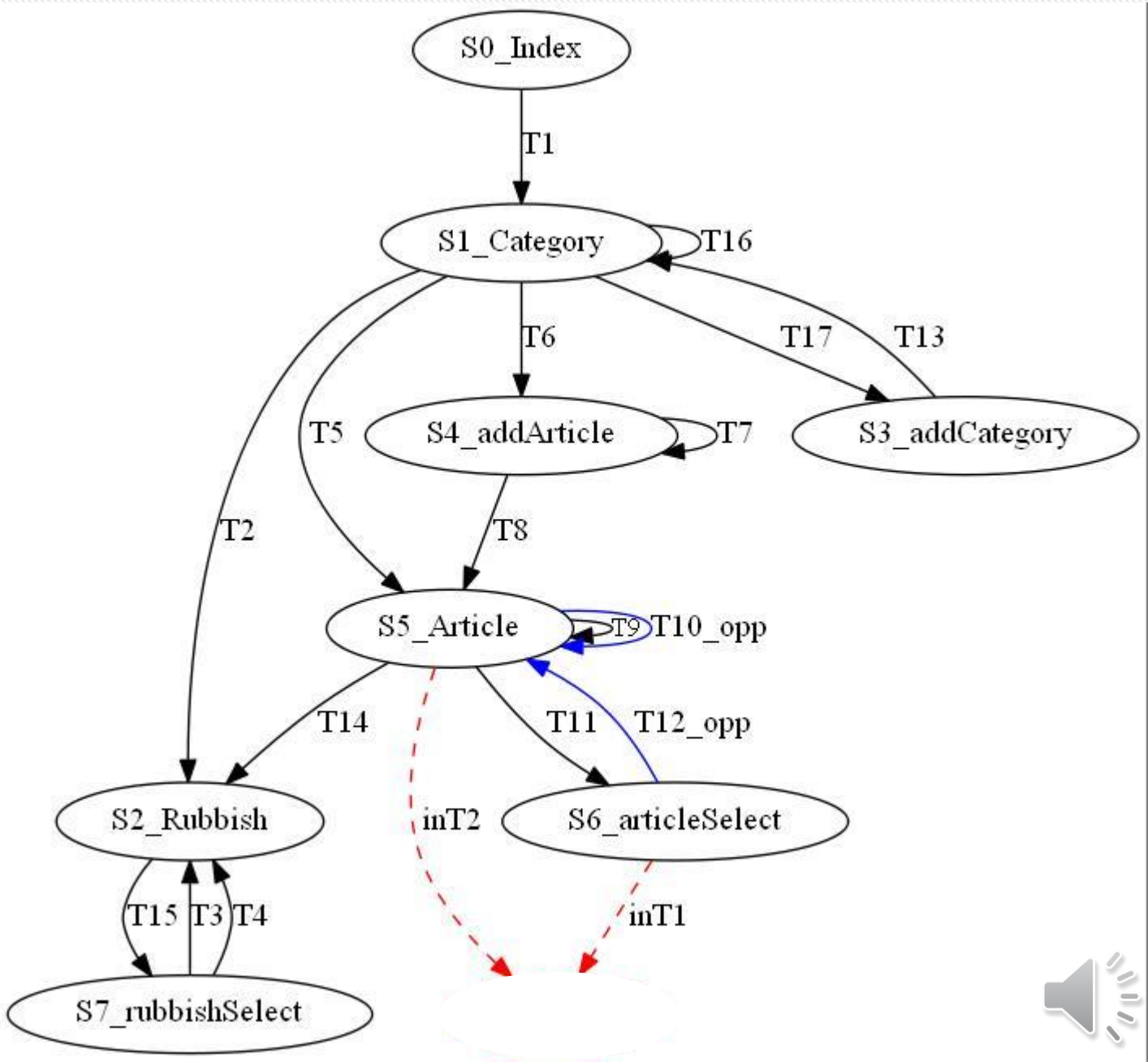
Case Study

- In the incomplete EFSM model, transitions covering opposite conditions are likely to be derived from the same state.



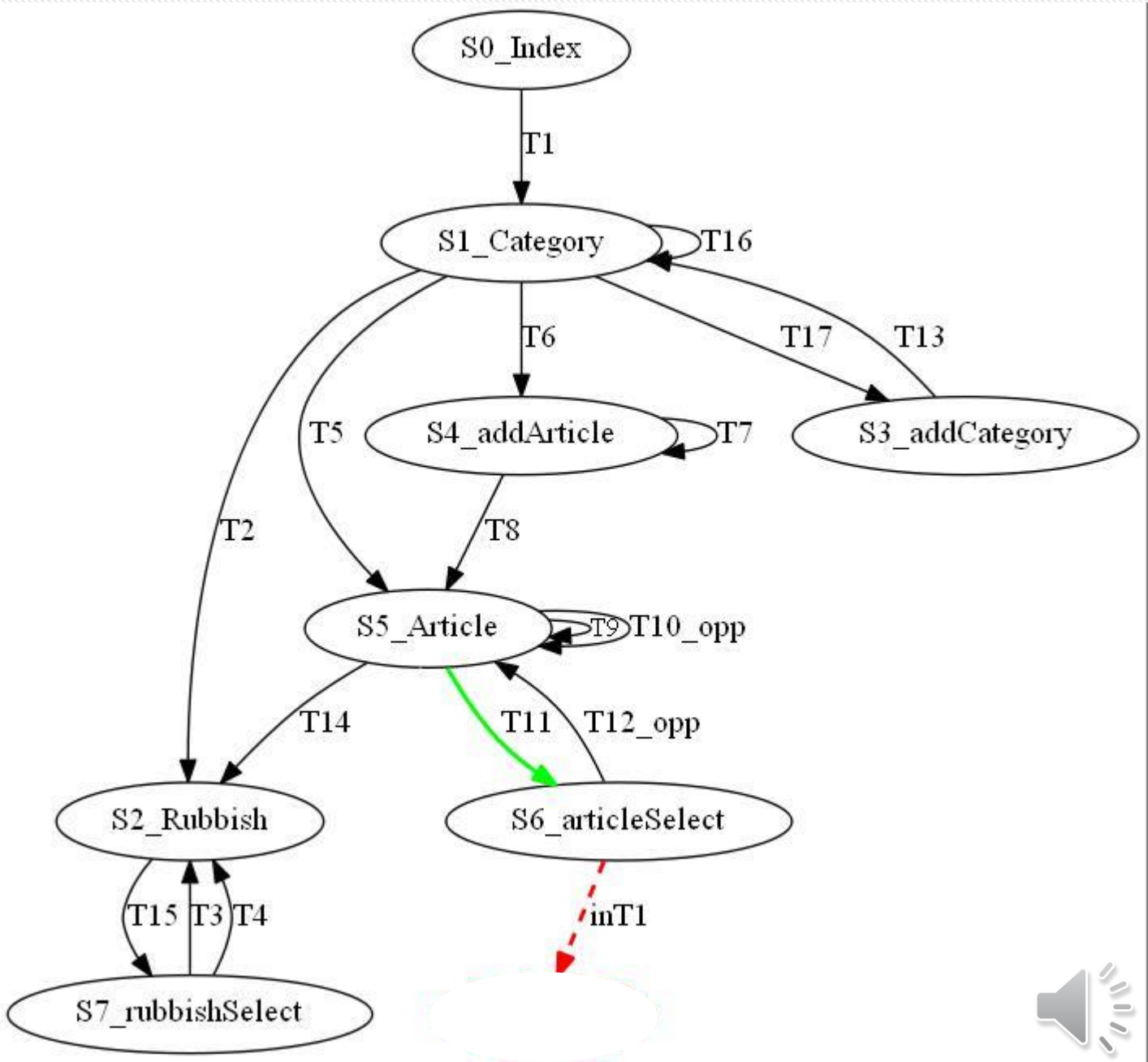
Case Study

- To complete the EFSM model in Figure, transitions inT1 or inT2 as well as their follow-up states should be supplemented into the model.



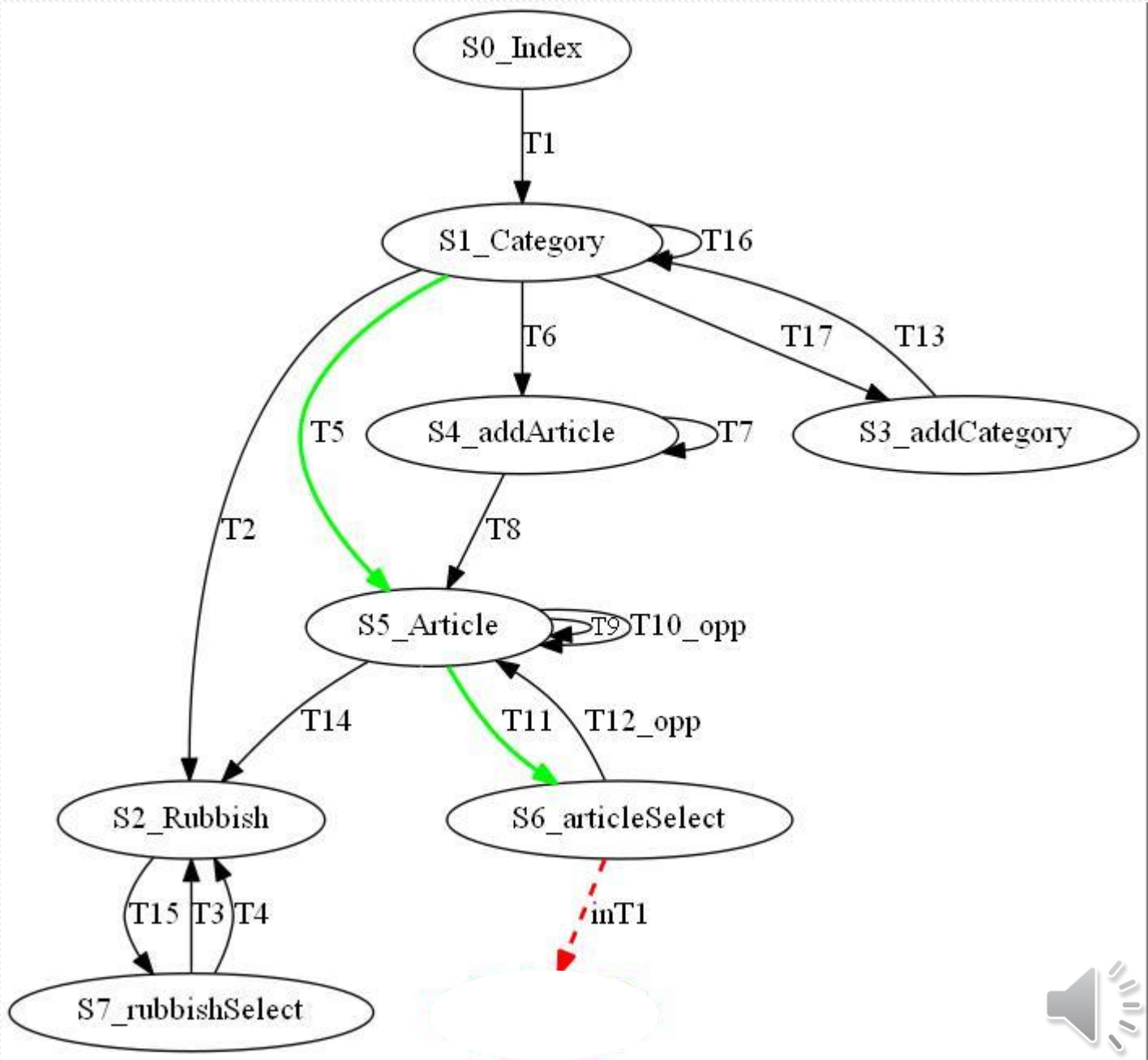
Case Study

- Which transitions (inT1 or inT2) should be considered first as the start for the sequence generation?
- The dependency between these two transitions and their preceding transitions can be analyzed.



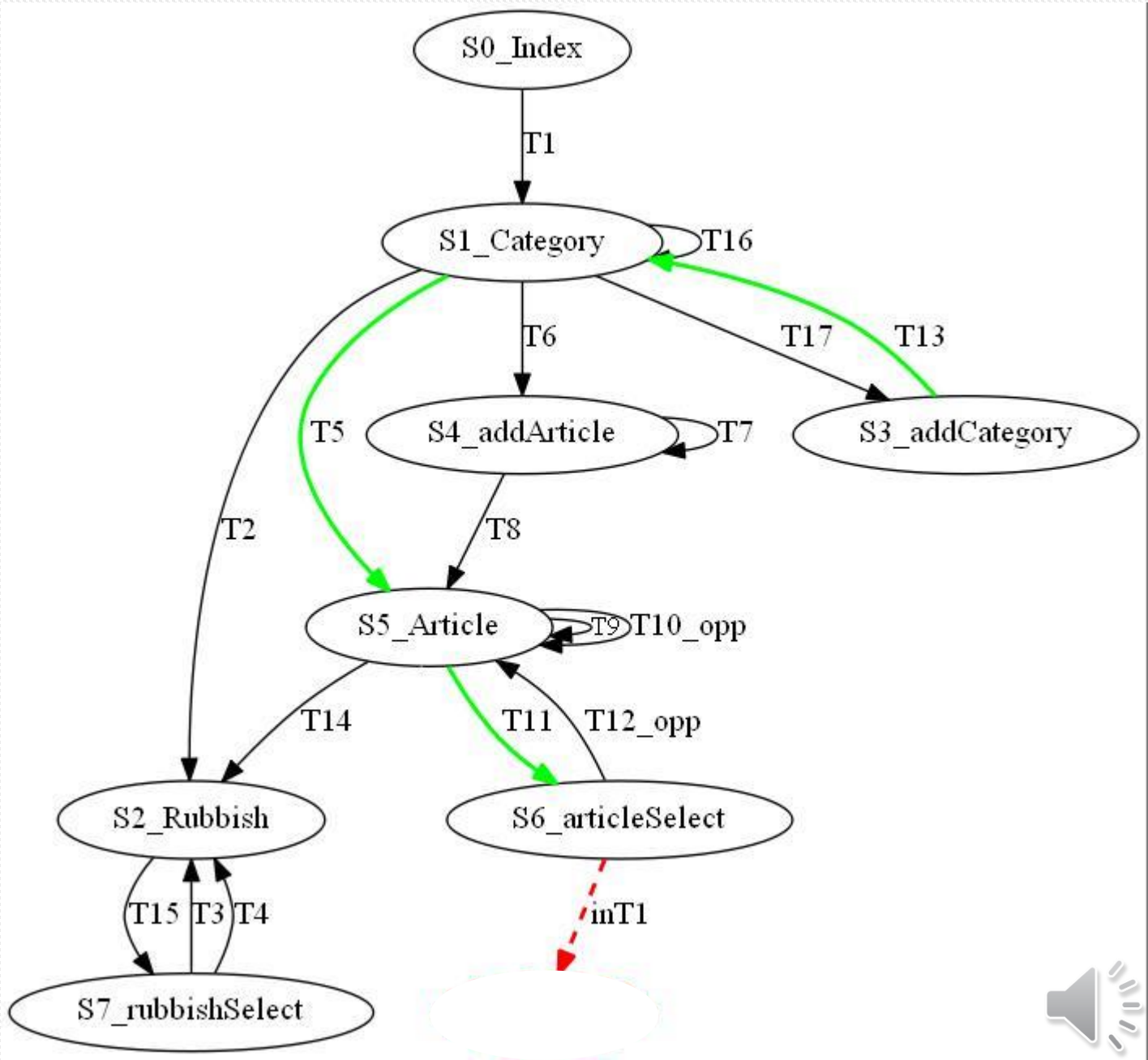
Case Study

- The less irrelevant variables are introduced by the preceding transitions, the less negative impact on sequence feasibility.



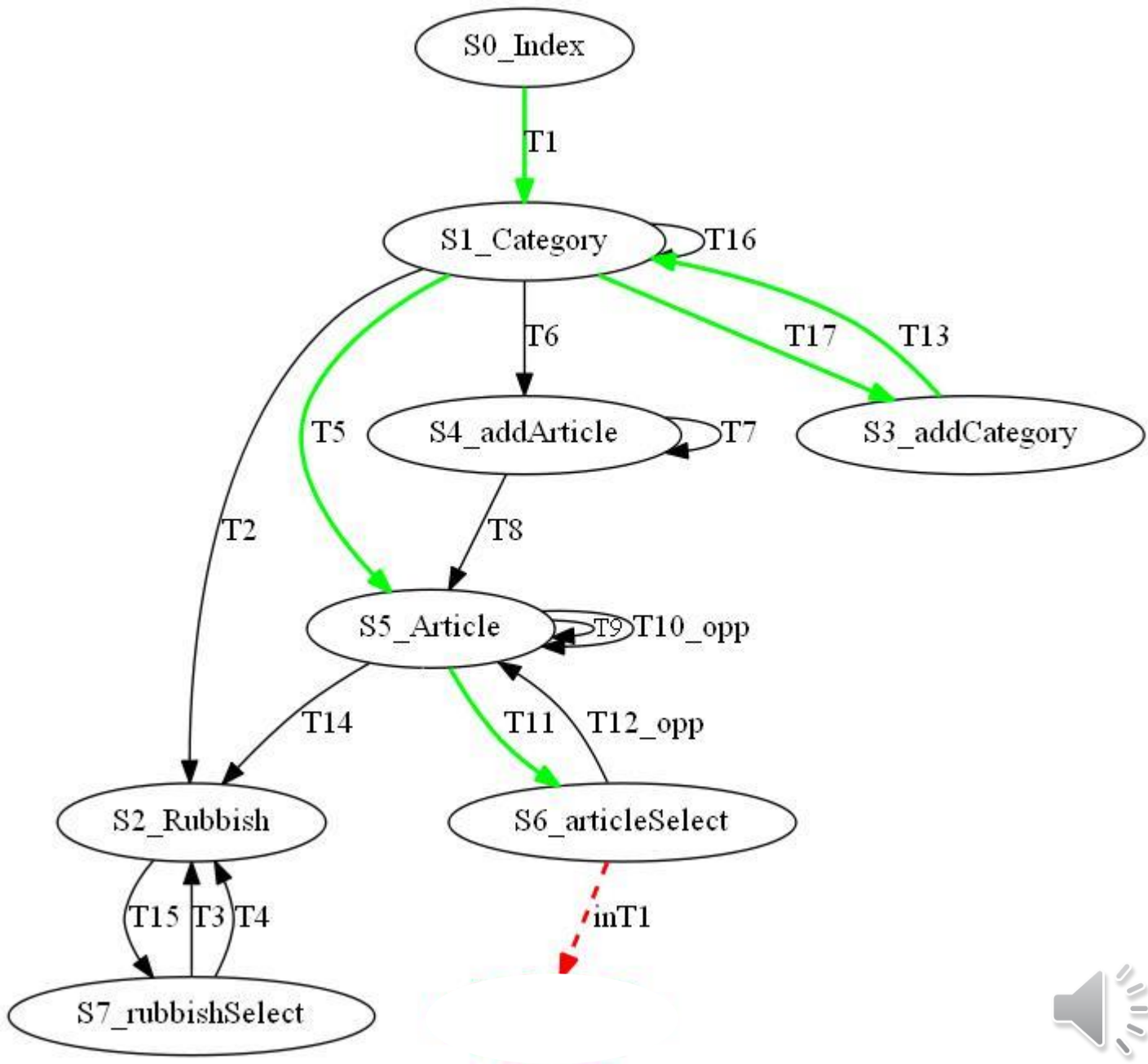
Case Study

- The more relevant variables defined by the preceding transitions, the more significant the positive impact on sequence feasibility.



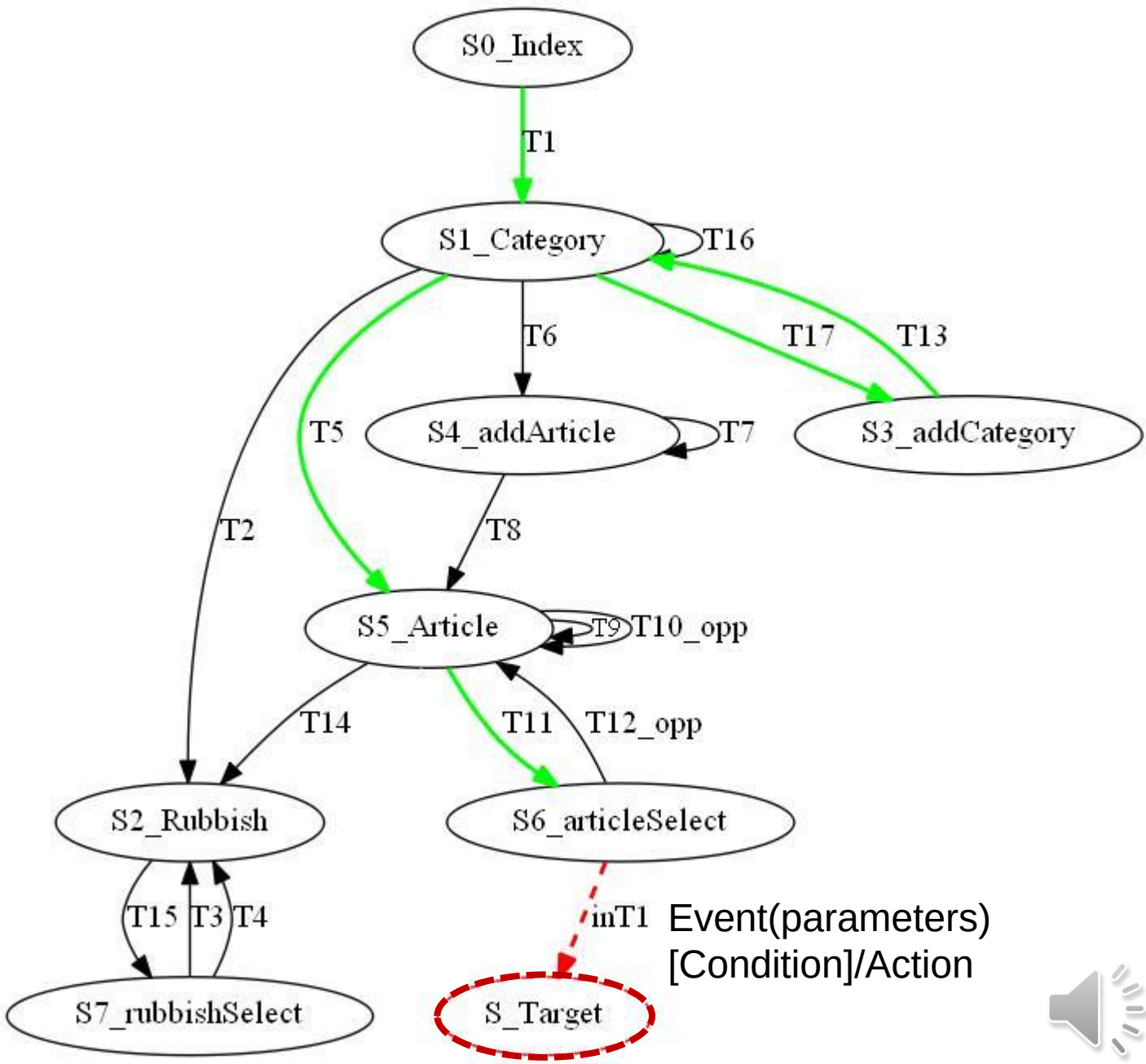
Case Study

- Feasible path generation



Case Study

- Model completion



PART TWO

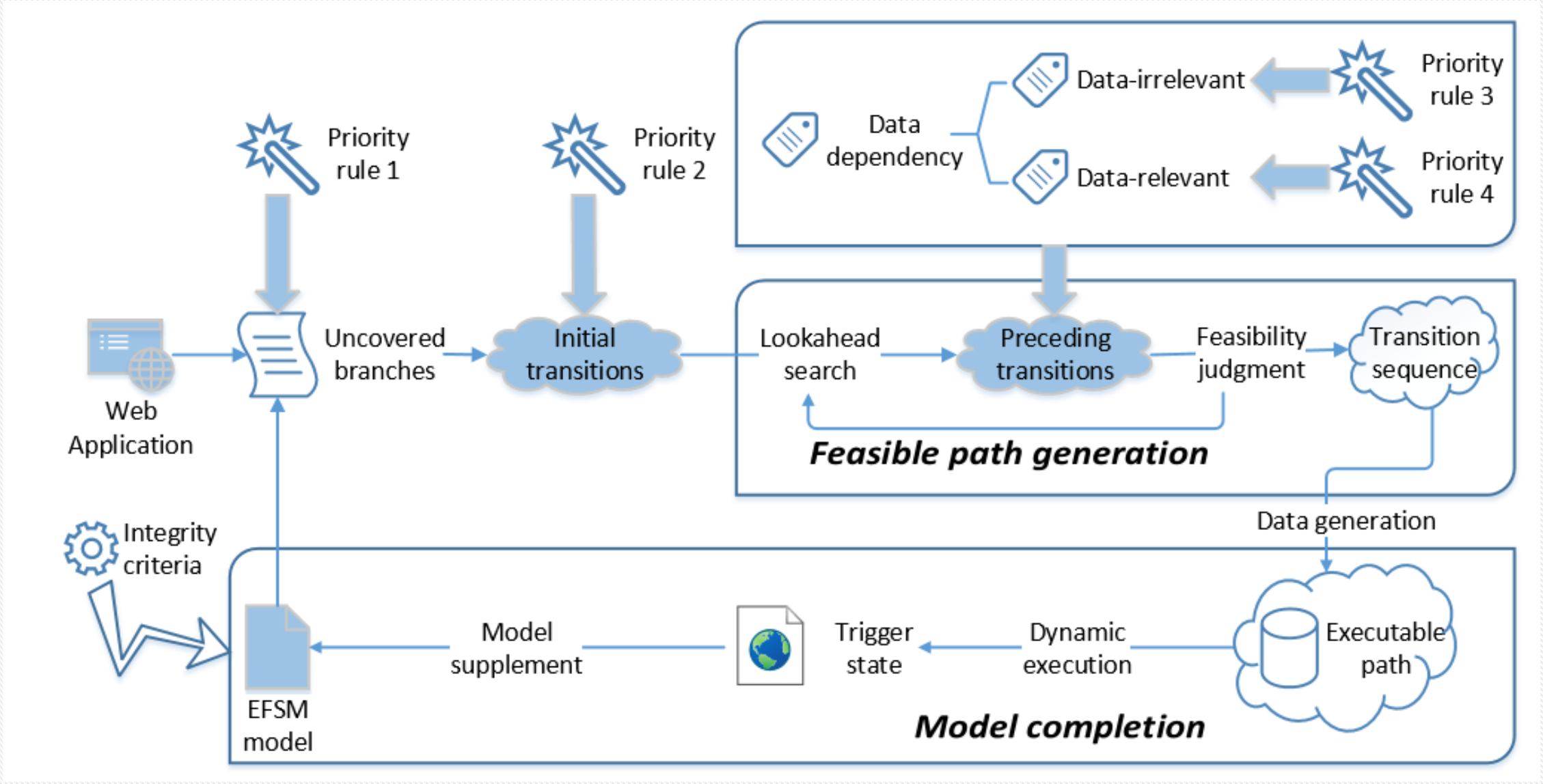
Approach

The Integrity Criterion for EFSM Model

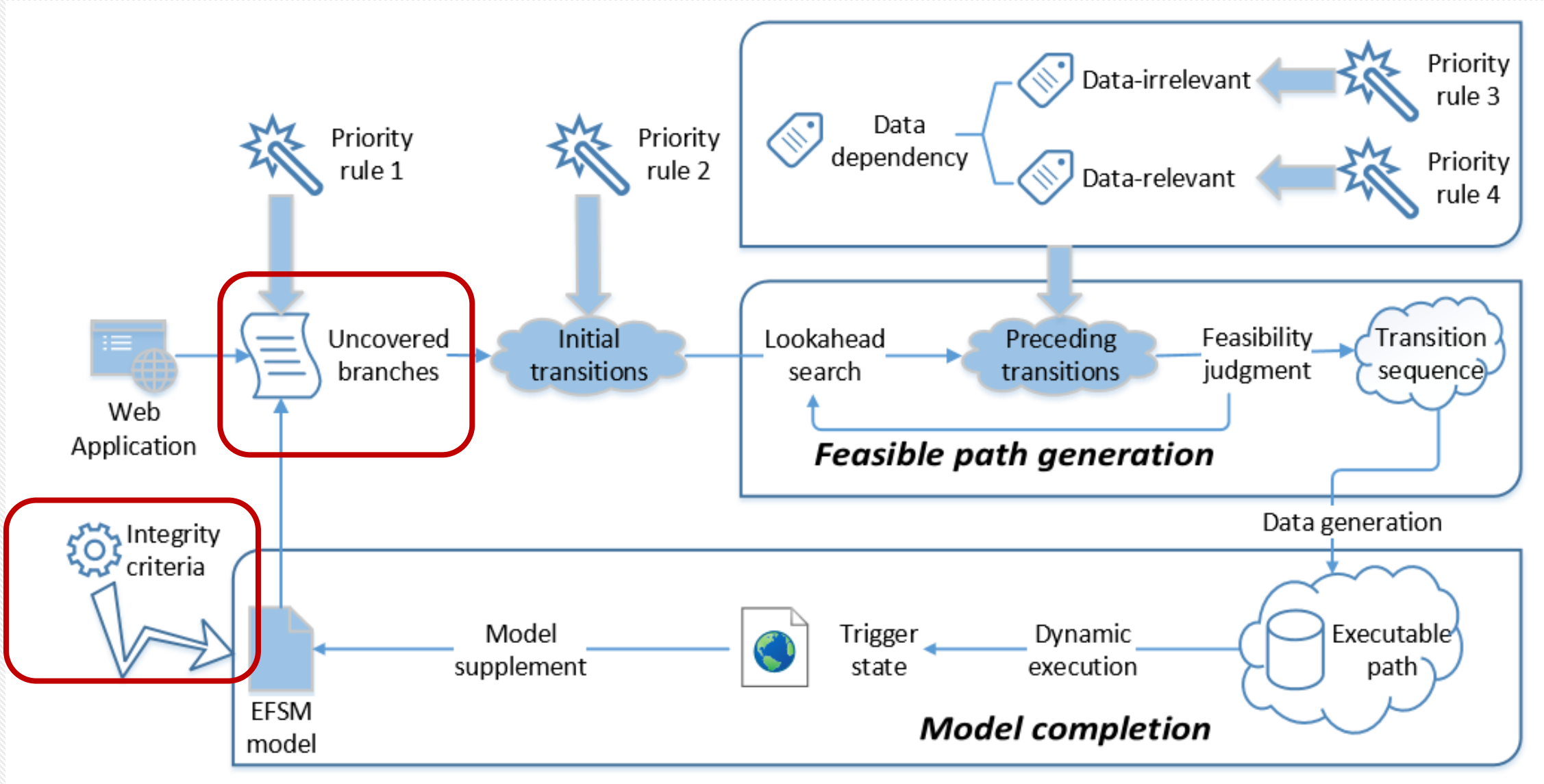
All events & JS branches coverage criterion

- If all the events and JS branches in the event handlers of a web application appear in its corresponding model, we call the model satisfying All events & JS branches coverage criterion.

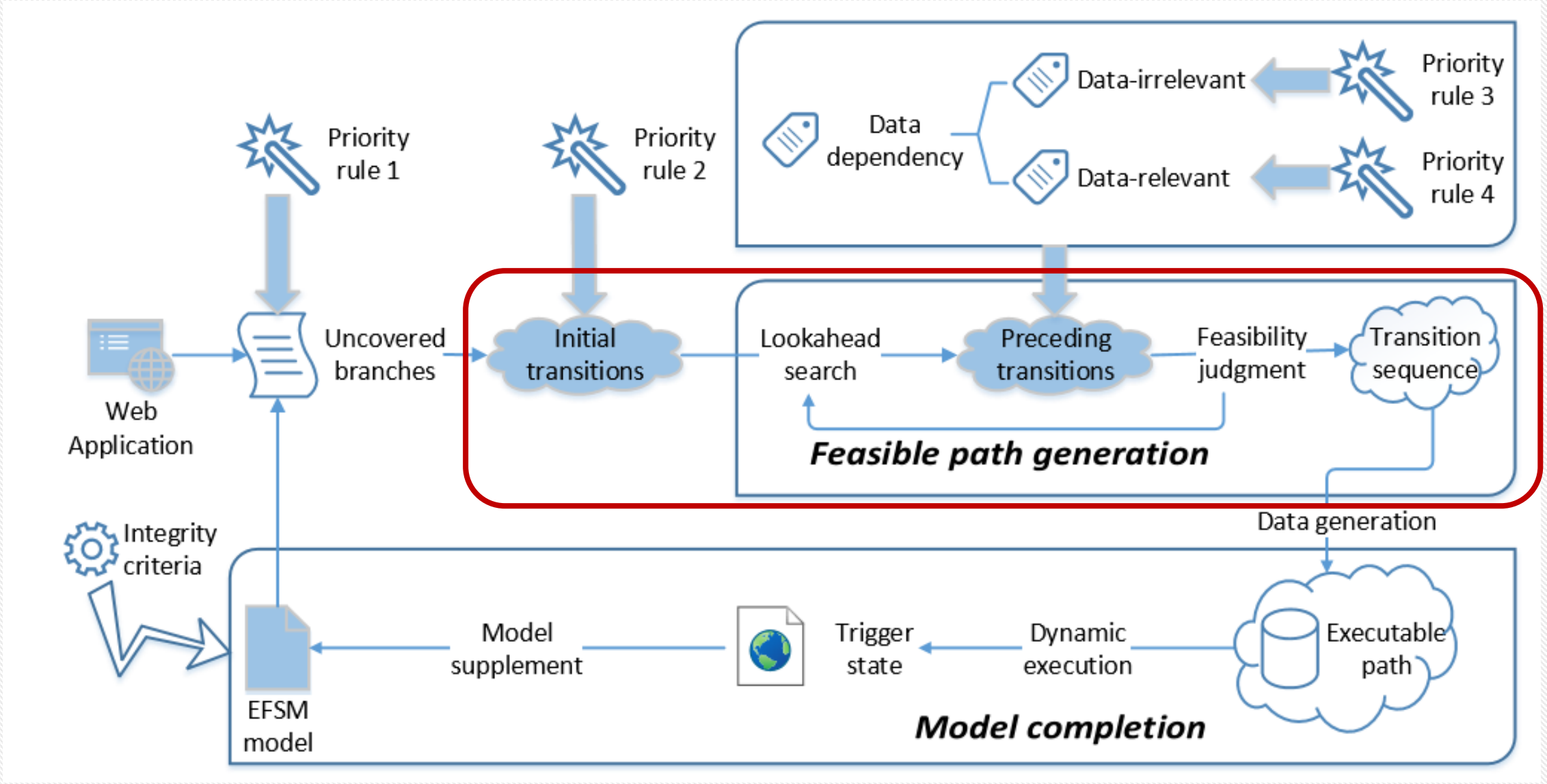
Model completion



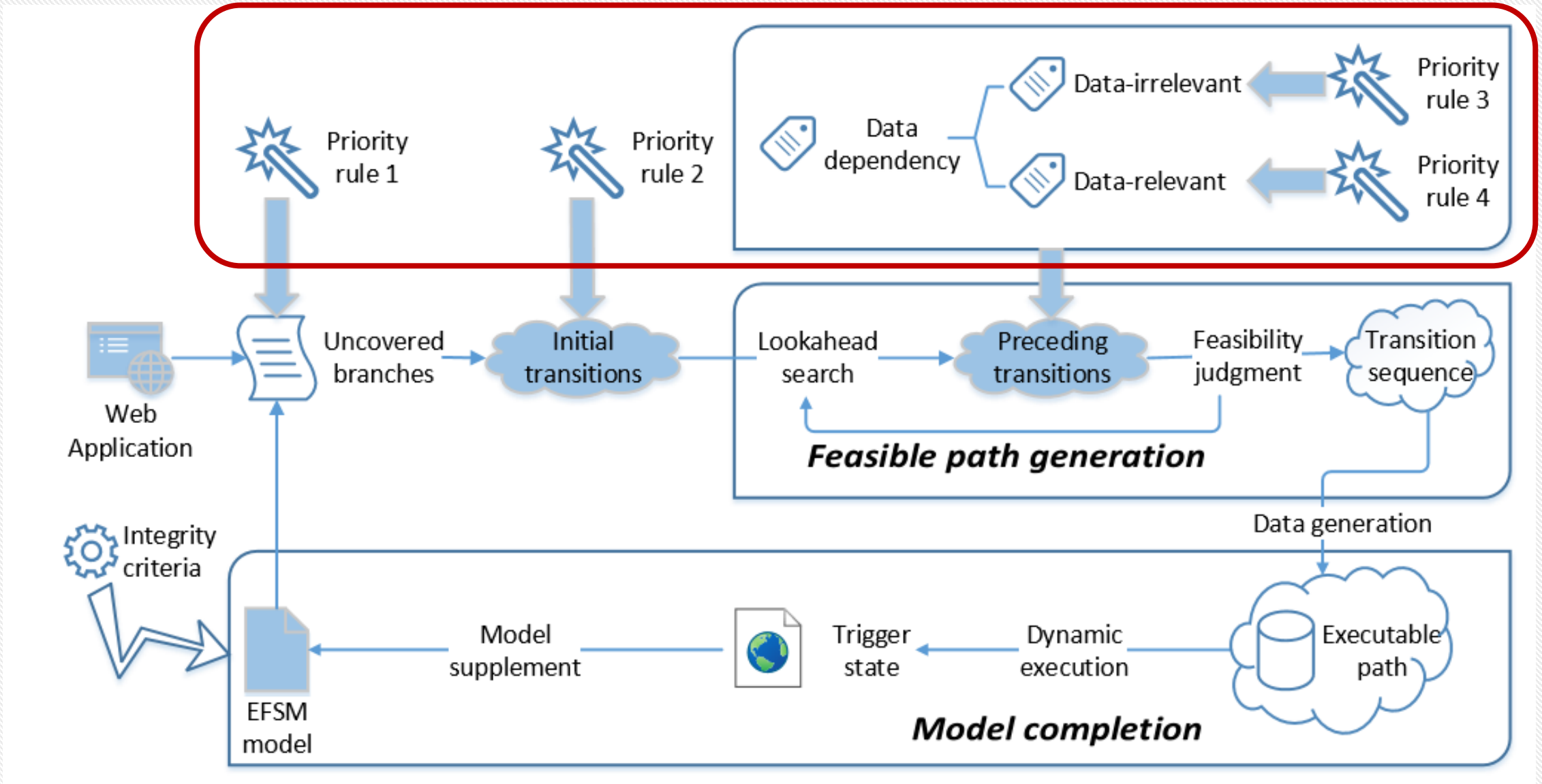
Model completion



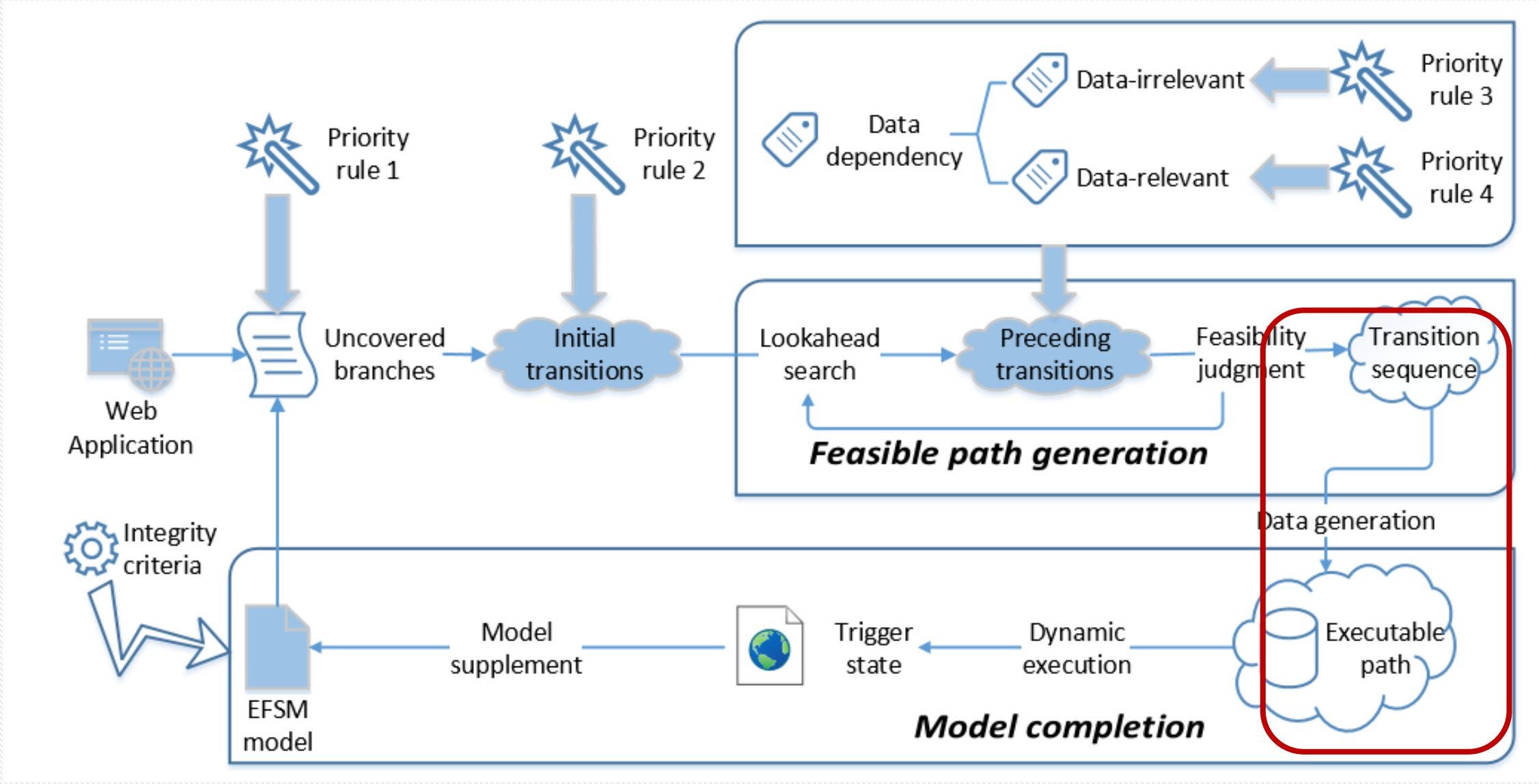
Model completion



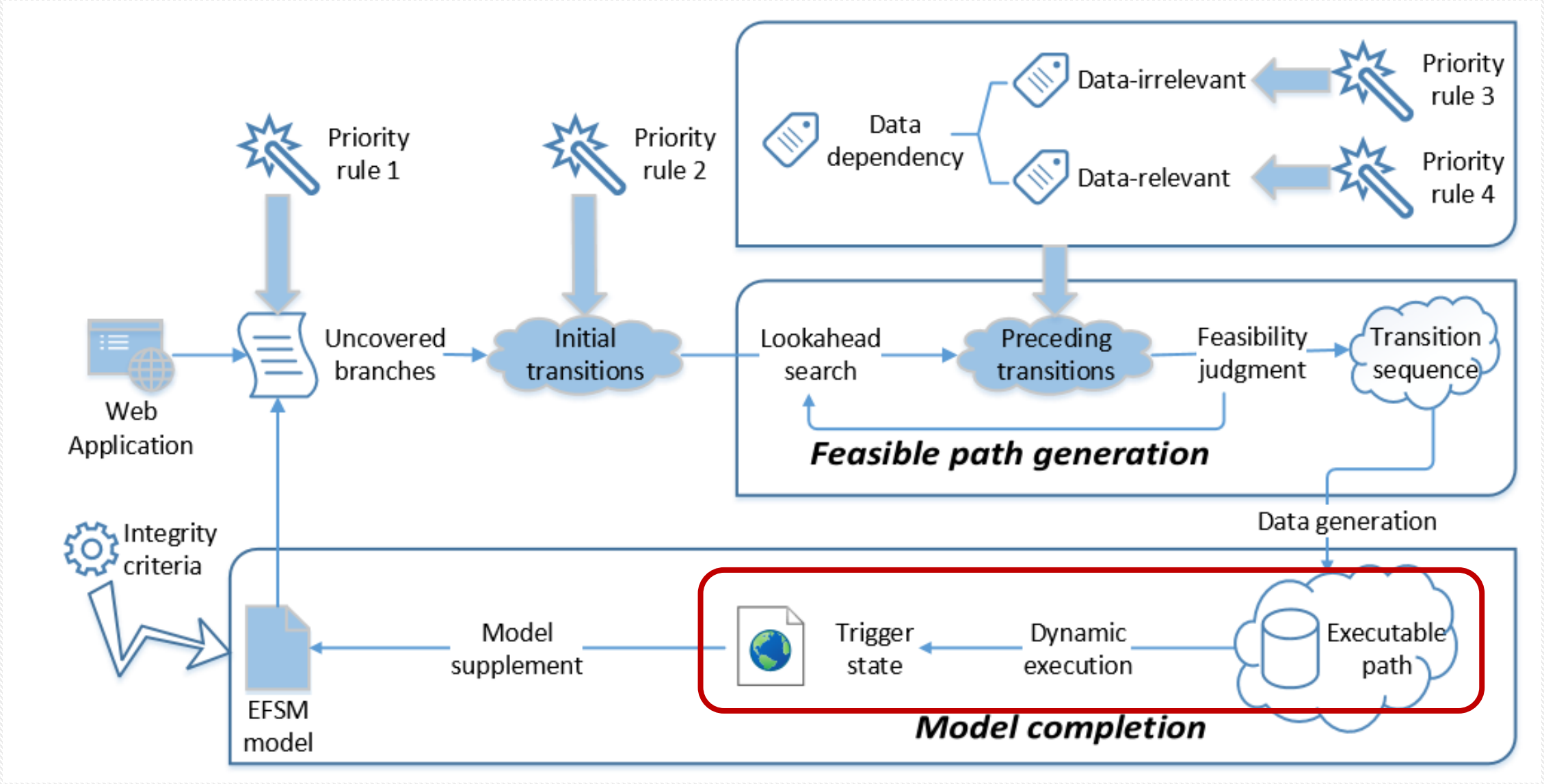
Model completion



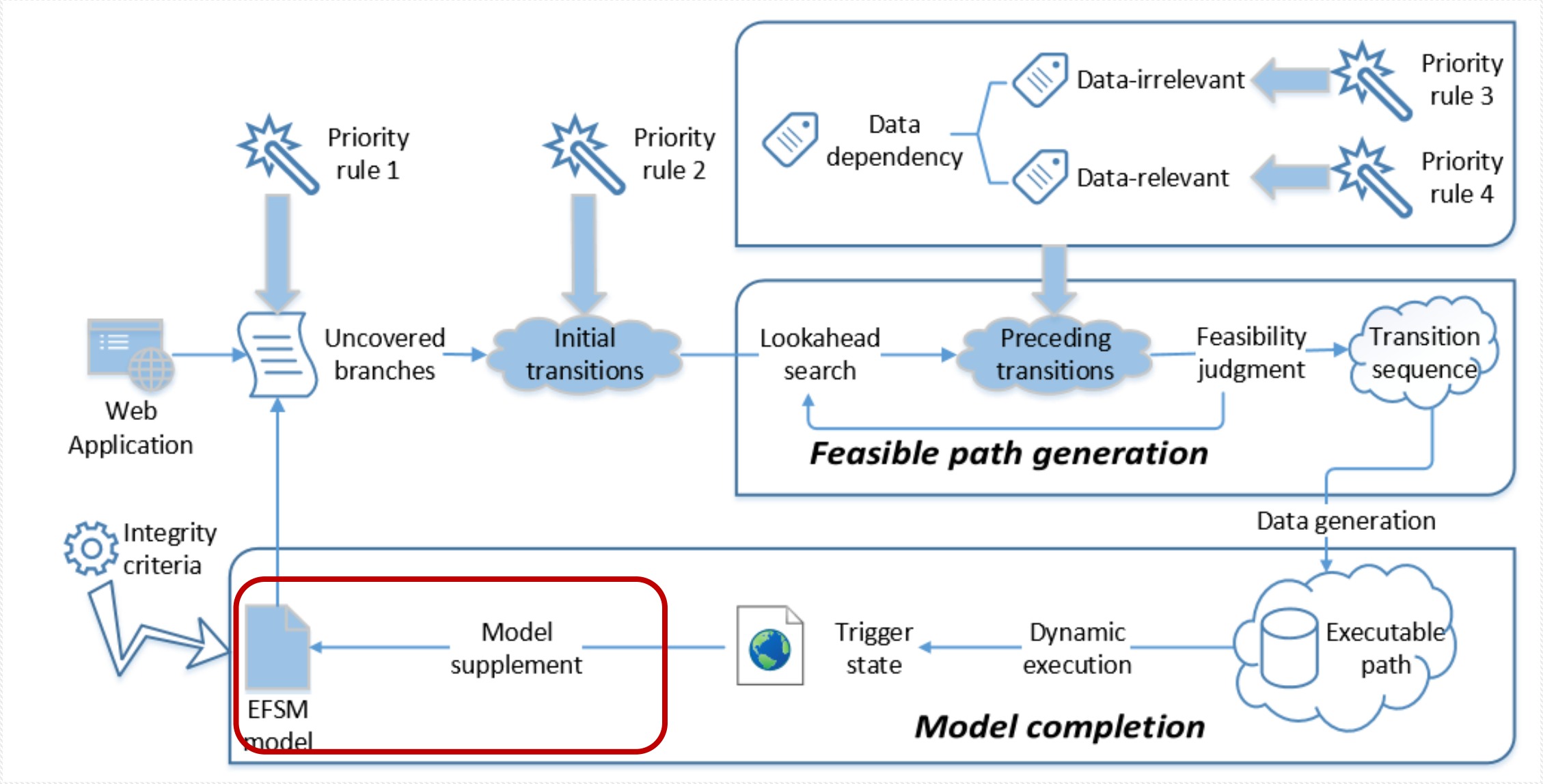
Model completion



Model completion



Model completion



PART THREE

Evaluation

Experiments

- LOC : the number of lines of code
- NJB : the number of all JS branches
- NUJB : the number of JS branches that are not covered by the EFSM models built based on user behavior traces

Table 1: Web applications used in the study

App Name	*LOC*	*NJB*	*NUJB*	Functional description
SchoolMate	8181	89	18	Student management system with admin role
Addressbook	47481	25	5	Addressbook management system
Webchess	4722	25	8	Online chess game
FAQForge	1712	8	3	FAQ management tool
phpaaCMS	15949	61	7	Article management system

Experiments

- RQ1 : Can the transition sequence generation method generate feasible transition sequences to traverse the uncovered behavior?

Table 2: The results of transition sequence generation

	SchoolMate	Addressbook	Webchess	FAQForge	phpaaCMS
NUJB	18	5	8	3	7
NTJB	18	5	7	3	7

- NTJB : the number of traversed branches by feasible transition sequences generated

Experiments

- RQ2 : Is our model completion method based on the feasible transition sequence effective in improving the model integrity?

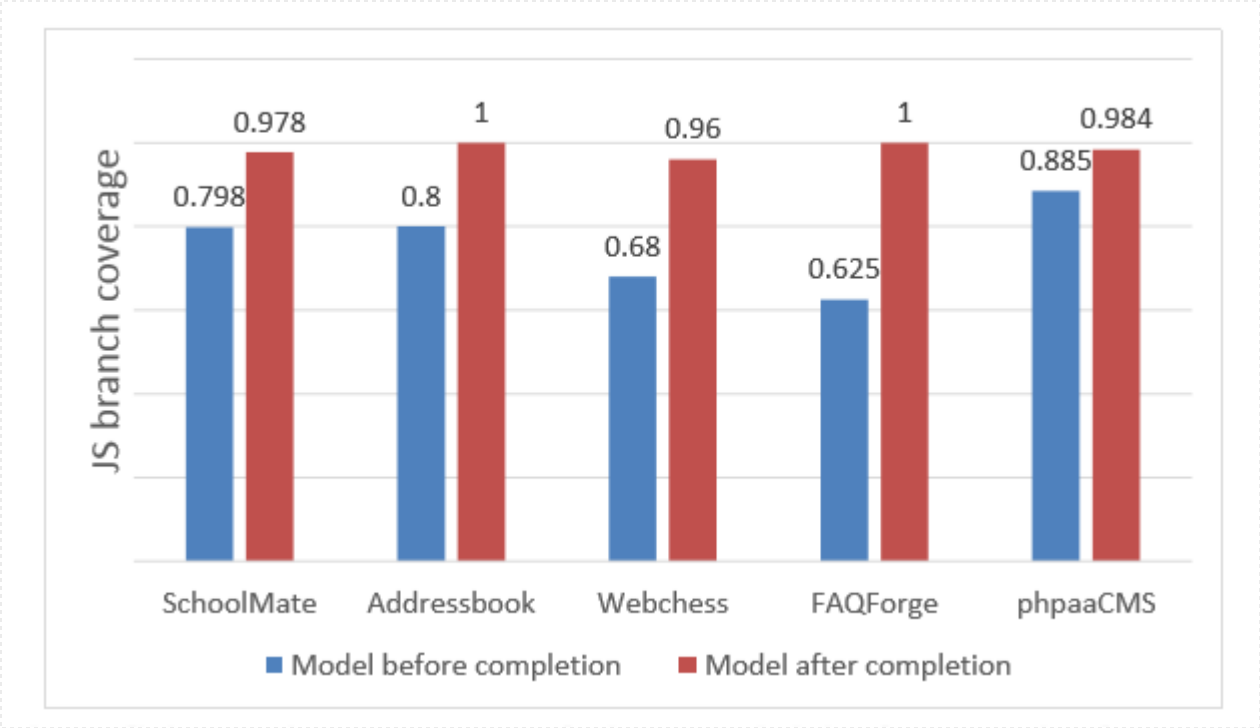
Table 3: The models before and after model completion

		SchoolMate	Addressbook	Webchess	FAQForge	phpaaCMS
$NTJB$		18	5	7	3	7
$NSJB$		16	5	7	3	6
Model before completion	NS	41	19	13	10	50
	NT	118	33	30	17	83
Model after completion	NS	45	20	14	11	54
	NT	140	38	39	22	91

- NSJB : the number of branches that are successfully supplemented to the model
- NS : the number of states
- NT : the number of transitions

Experiments

- RQ2 : Is our model completion method based on the feasible transition sequence effective in improving the model integrity?



The JS branch coverage of the model before and after completion

Experiments

- RQ3 : How effective are the priority rules in transition sequence generation method?

LS	lookahead search without priority rules for preceding transitions
AutoMC	lookahead search with all priority rules
AutoMC-DR	lookahead search with all priority rules except for data-relevant transitions
AutoMC-DI	lookahead search with all priority rules except for data-irrelevant transitions

Experiments

- RQ3 : How effective are the priority rules in transition sequence generation method?

- NTS : the total number of transition sequences generated
- AGT(ms) : the average generation time of each feasible transition sequence
- ALS : the average length of the feasible transition sequences

Experiments

- RQ3 : How effective are the priority rules in transition sequence generation method?

Table 4: The results of transition sequence generation by different methods

Method		SchoolMate	Addressbook	Webchess	FAQForge	phpaaCMS
LS	*NTJB*	18	5	8	3	7
	NTS	267	451	680	256	496
	AGT	36.93	15.24	187.14	12.10	4.29
	ALS	21.80	24.39	145.07	16.23	8.99
AutoMC-DR	*NTJB*	18	5	8	3	7
	NTS	275	455	689	258	518
	AGT	18.89	13.90	183.30	11.17	4.39
	ALS	16.78	23.02	135.78	14.89	8.98
AutoMC-DI	*NTJB*	18	5	7	3	7
	NTS	1744	478	606	294	521
	AGT	7.88	7.51	28.12	2.59	2.72
	ALS	12.09	19.82	30.47	12.87	8.99
AutoMC	*NTJB*	18	5	7	3	7
	NTS	1743	483	618	293	520
	AGT	8.02	6.70	25.09	2.37	3.03
	ALS	12.08	18.95	29.56	11.50	8.95

Conclusion

- We define an integrity criterion to evaluate completeness of EFSM model for web applications.

- We propose an automatic model completion method based on static analysis and dynamic execution for EFSM models of web applications.
  - Feasible transition sequence generation for target transition.
  - Model completion based on feasible transition sequence.

Conclusion

- The experimental results show that the proposed method can generate feasible transition sequences for uncovered branches and complete the EFSM model built based on user behavior traces.

Conclusion

- The experimental results show that the proposed method can generate feasible transition sequences for uncovered branches and complete the EFSM model built based on user behavior traces.

- The integrity of the model has been significantly improved. The average JS branch coverage of the model increased from 75.76% to 98.44%.

Thanks