Toward a UCM-based Approach for Recovering System Availability Requirements from Execution Traces

Jameleddine Hassine
King Fahd University of Petroleum and Minerals,
Dhahran, Saudi Arabia
jhassine@kfupm.edu.sa

Wahab Hamou-Lhadj
Software Behaviour Research (SBA) Lab
ECE, Concordia University, Montreal, Canada
abdelw@ece.concordia.ca

Goal and Applications

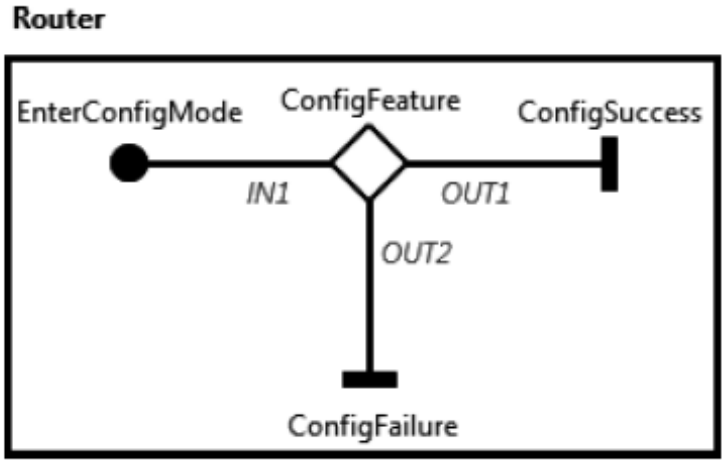
- **Goal:**
  - To provide an approach based on UCM, a high-level visual requirement description language, for recovering system availability features from execution logs.

- **Applications:**
  - Verification of system implementation w.r.t. availability requirements
  - Understanding, analyzing, and system debugging
  - Simulation of system scenarios
  - Documentation and knowledge sharing

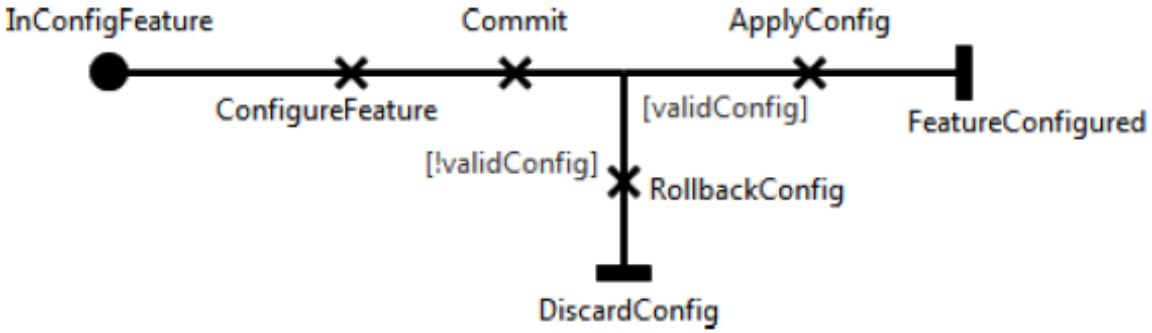
UCM: Use Case Maps

- Part of the ITU-T User Requirements Notation (URN) standard.
- A high-level visual scenario-based modeling language.
- Used to capture and integrate functional requirements in terms of causal scenarios.
- Static and behaviour of the system are shown in one diagram.
- Provides the stakeholders with guidance and reasoning about the system-wide architecture and behaviour.
- Good tool support: JUCMNav.

Example of a UCM

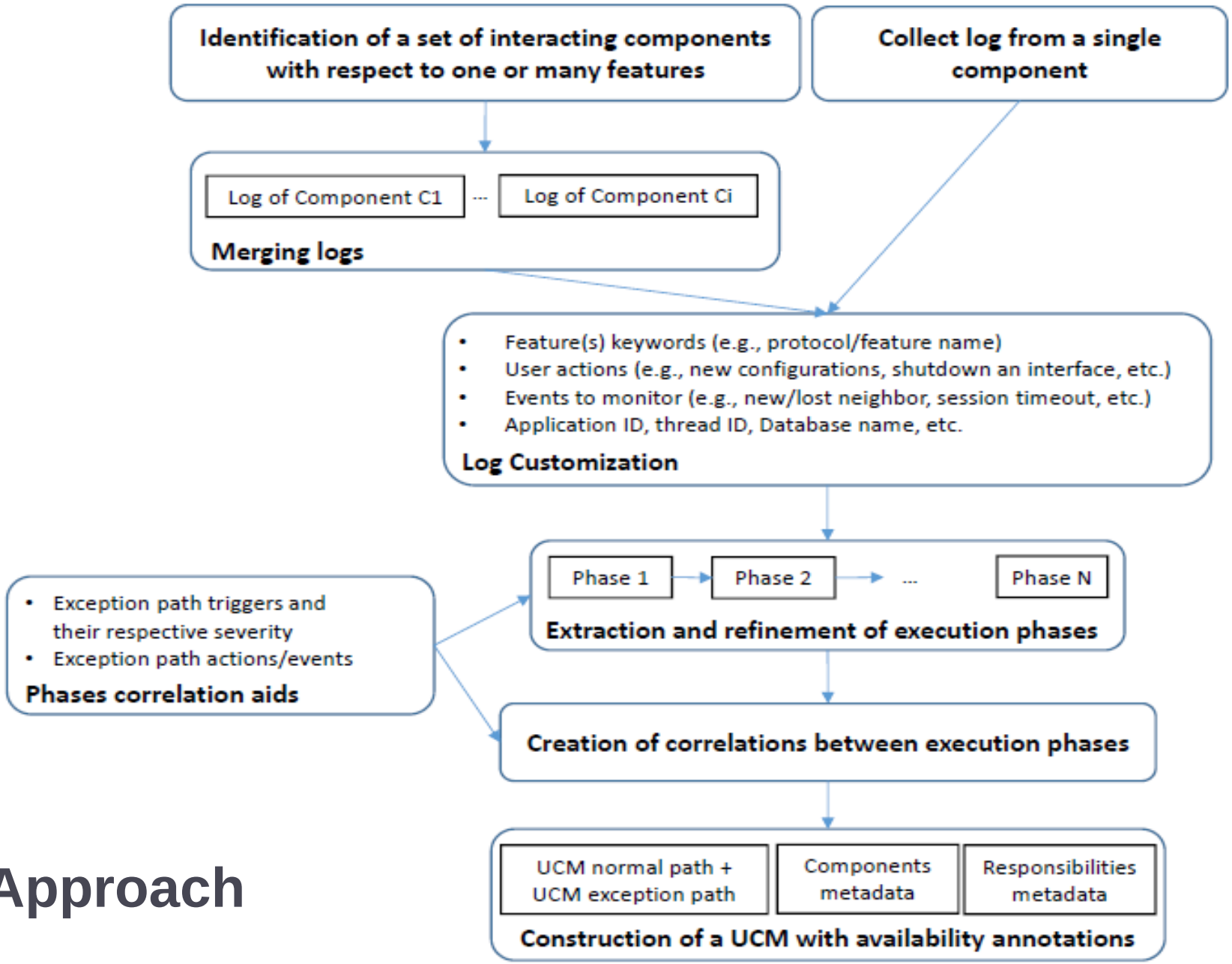


(a) Feature Configuration Scenario

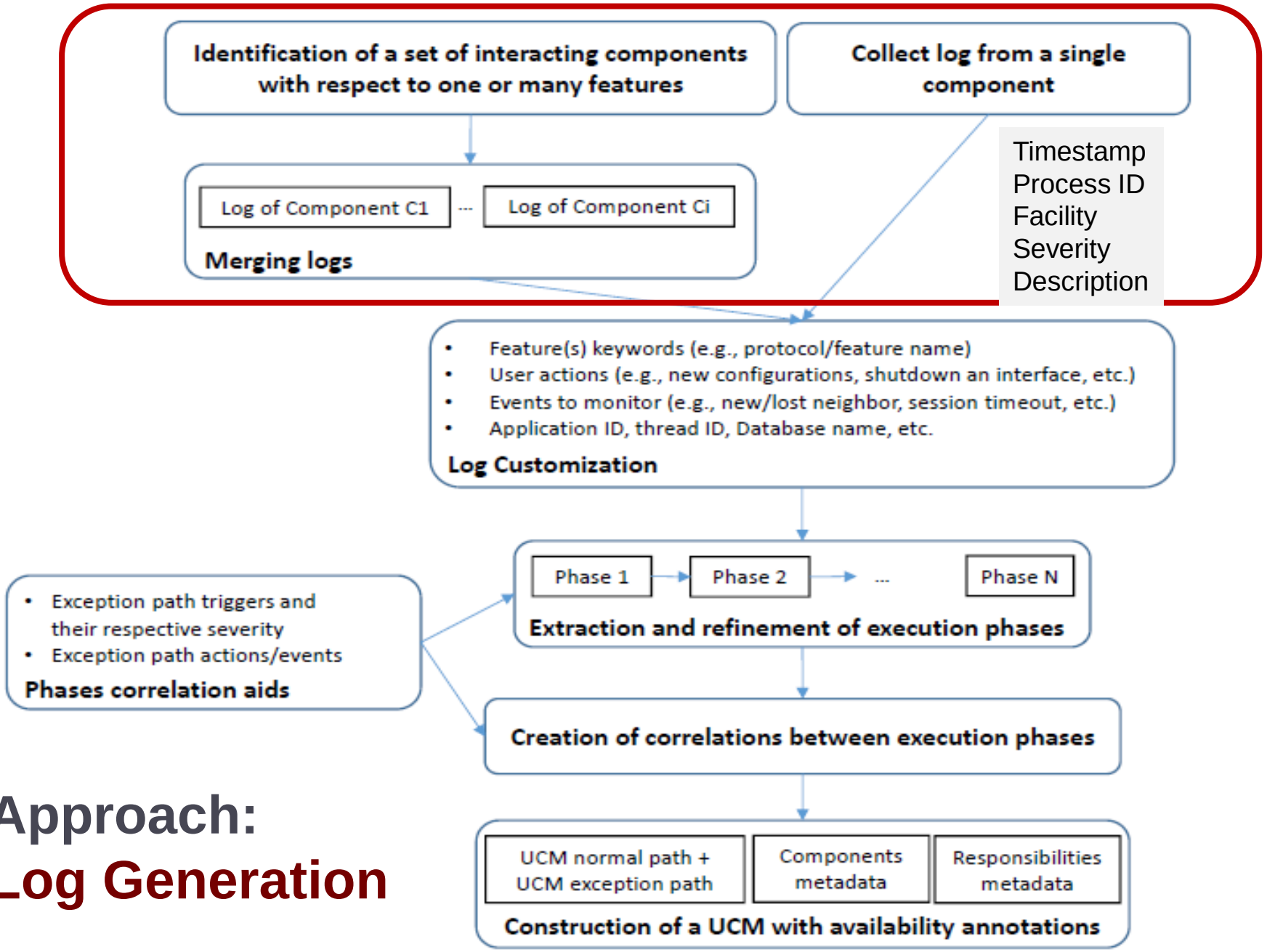


(b) FeatureConfig Stub Plugin

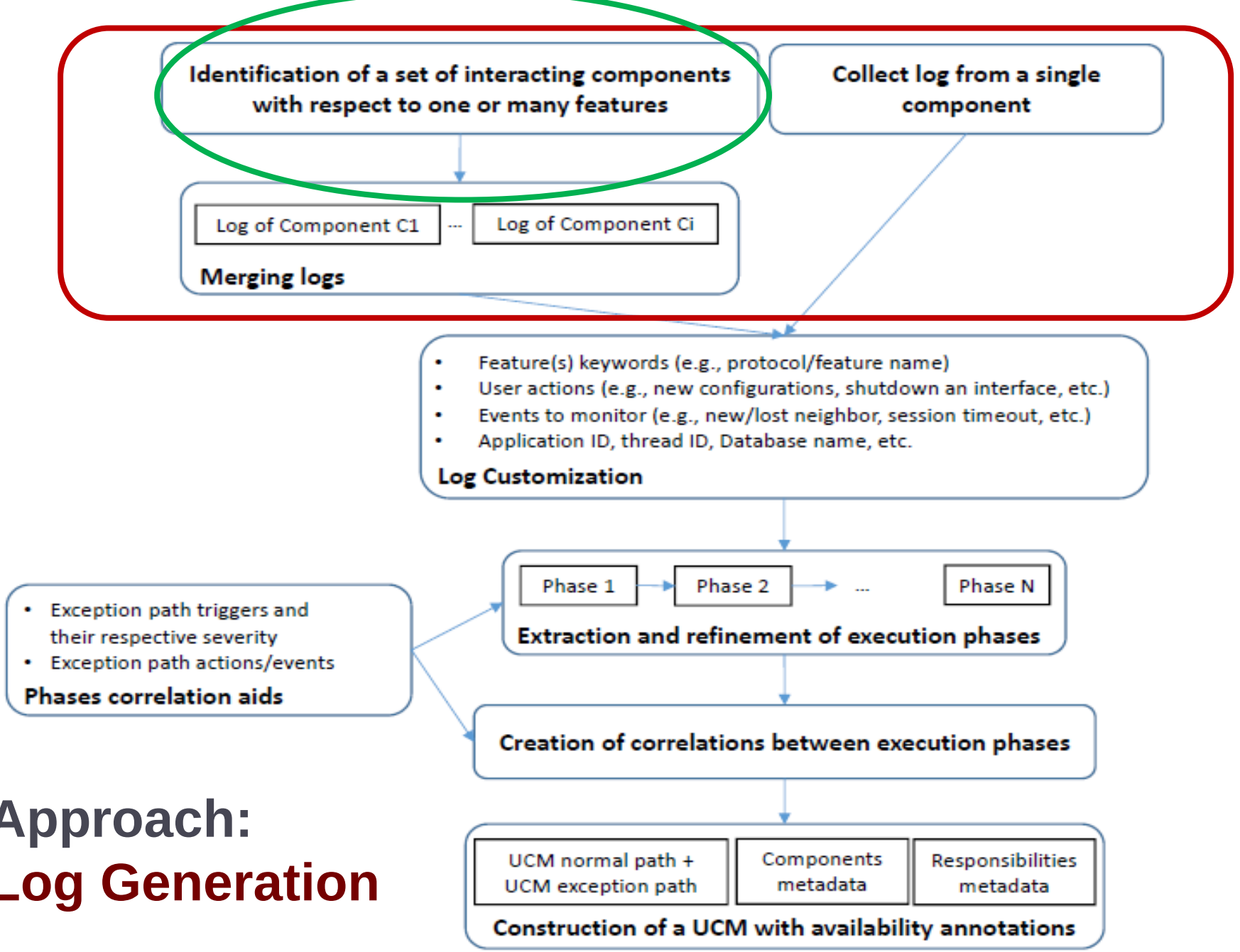
Approach


Identification of a set of interacting components with respect to one or many features
Collect log from a single component
Log of Component C1
...
Log of Component Ci
Merging logs
Feature(s) keywords (e.g., protocol/feature name)
User actions (e.g., new configurations, shutdown an interface, etc.)
Events to monitor (e.g., new/lost neighbor, session timeout, etc.)
Application ID, thread ID, Database name, etc.
Log Customization
Exception path triggers and their respective severity
Exception path actions/events
Phases correlation aids
Phase 1
Phase 2
...
Phase N
Extraction and refinement of execution phases
Creation of correlations between execution phases
UCM normal path + UCM exception path
Components metadata
Responsibilities metadata
Construction of a UCM with availability annotations

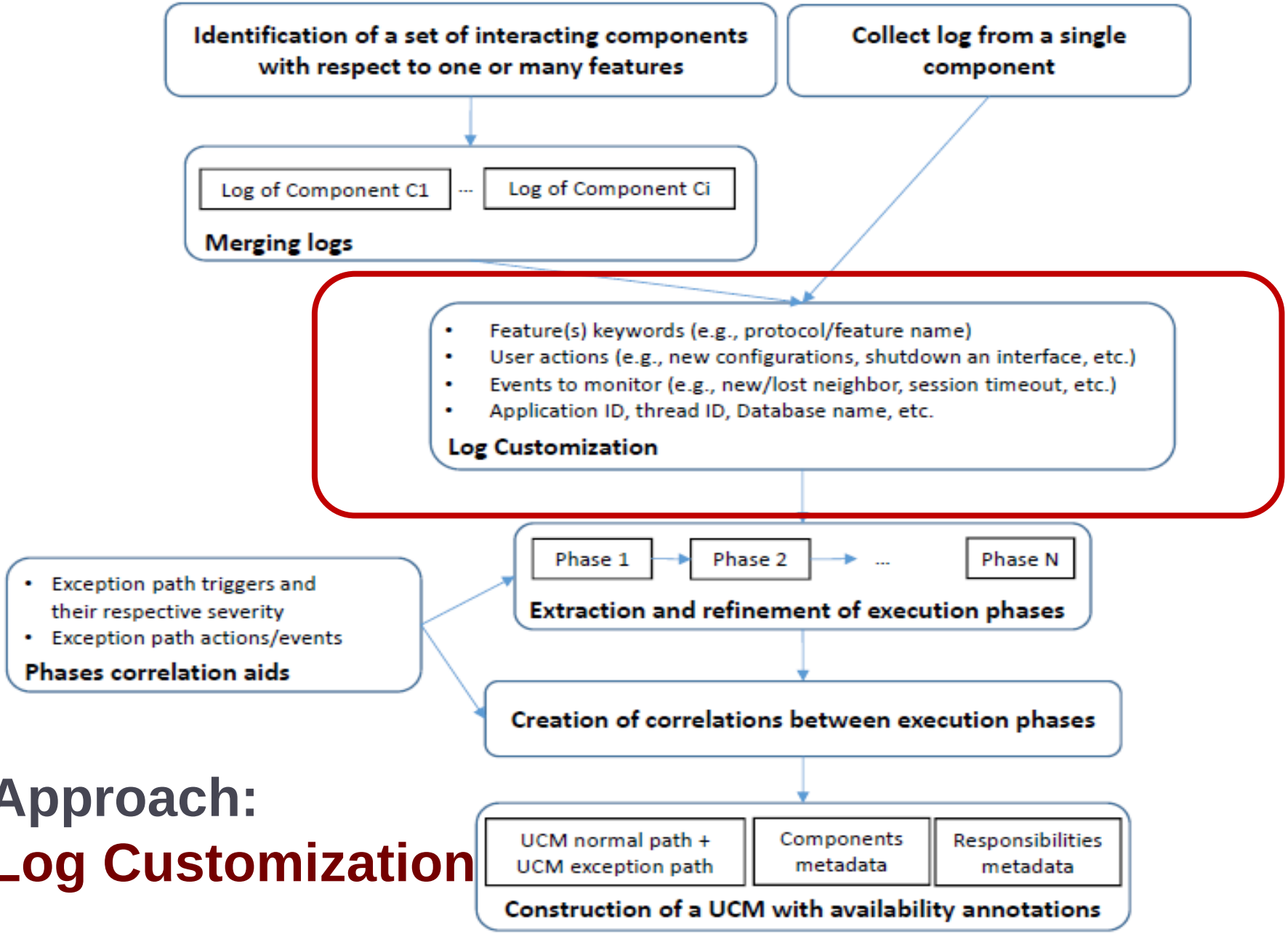
Approach:
Log Generation

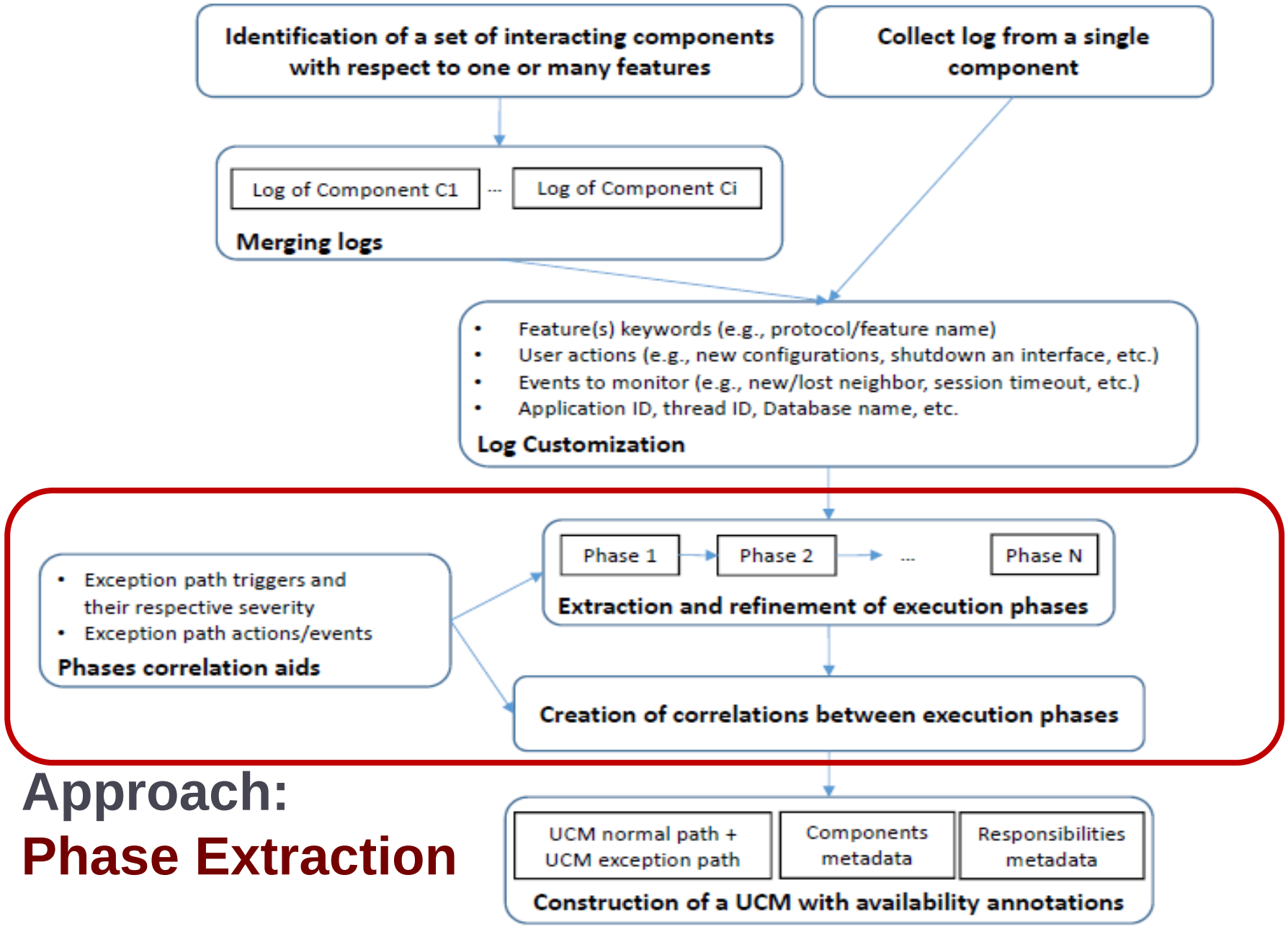


Approach:
Log Generation



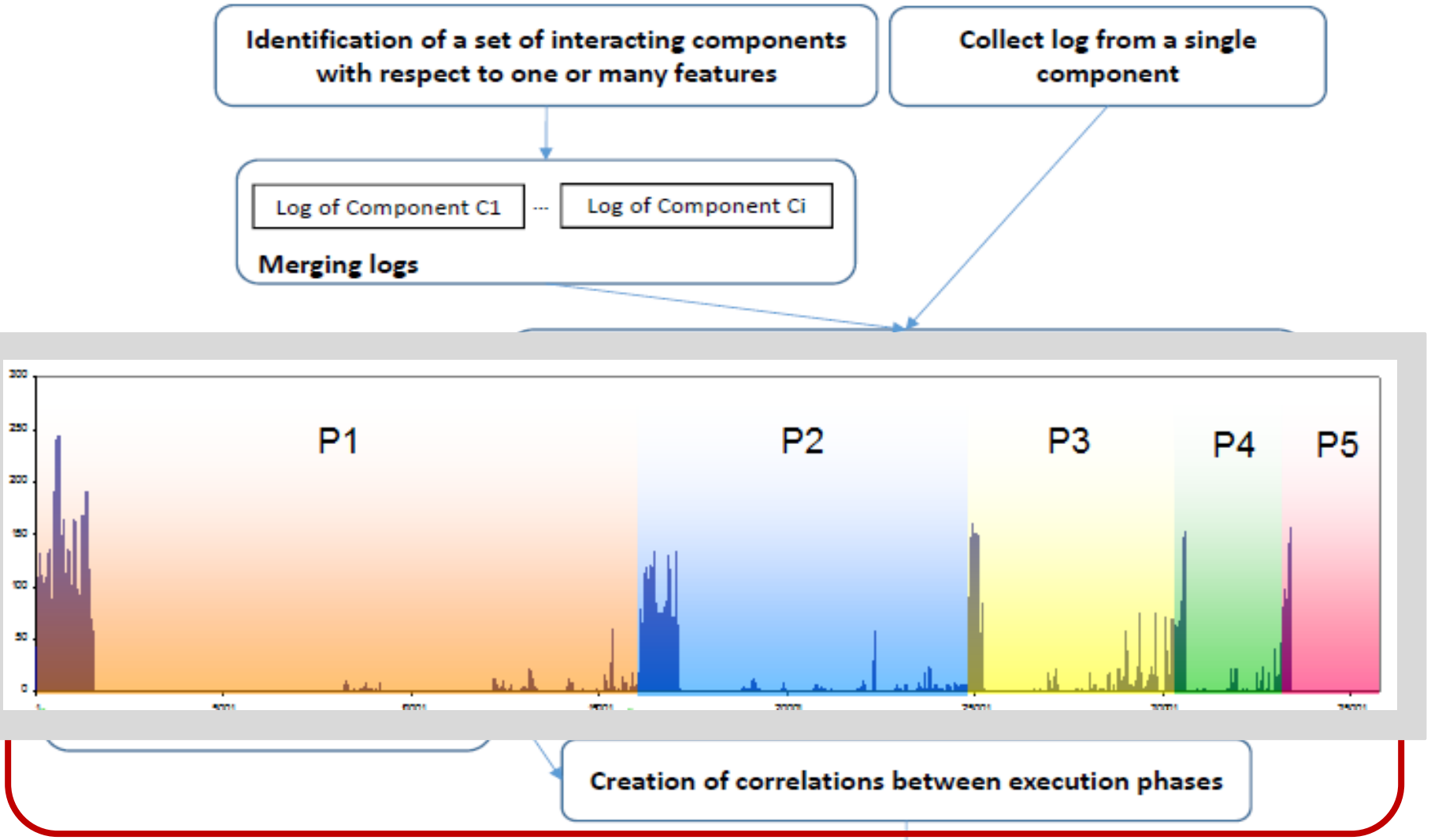
Approach:
Log Customization

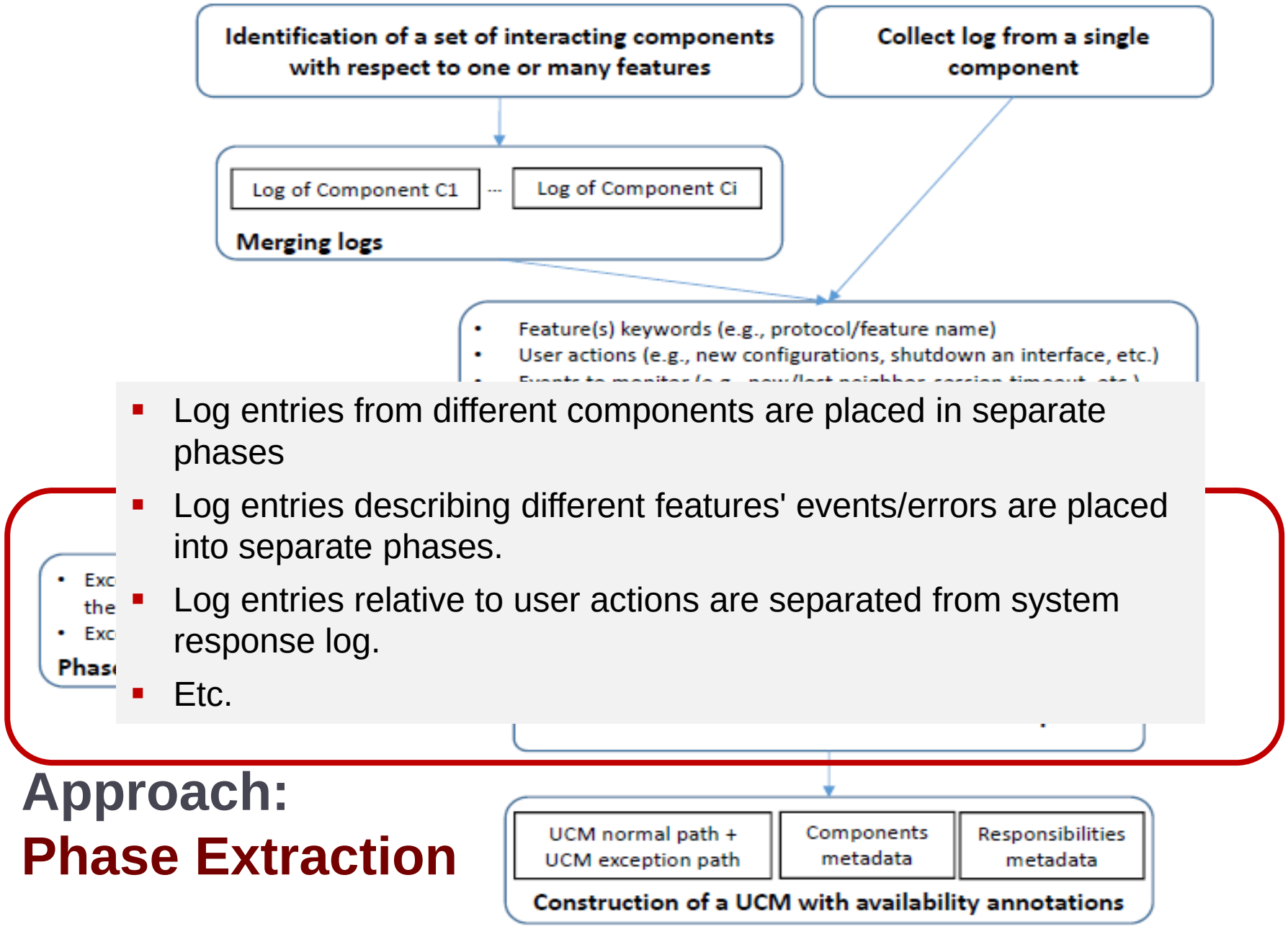




Approach:
Phase Extraction

Approach:
Phase Extraction

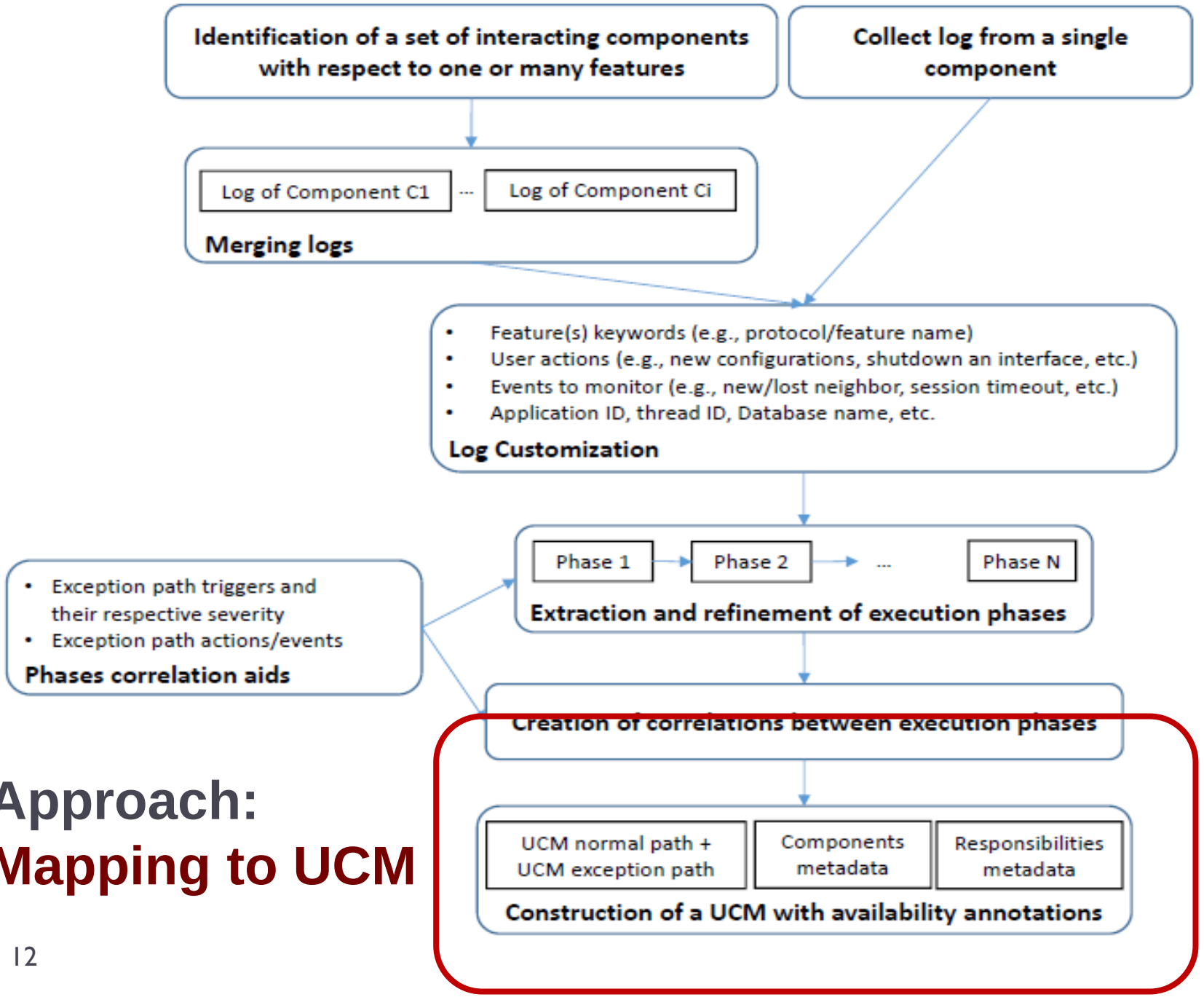


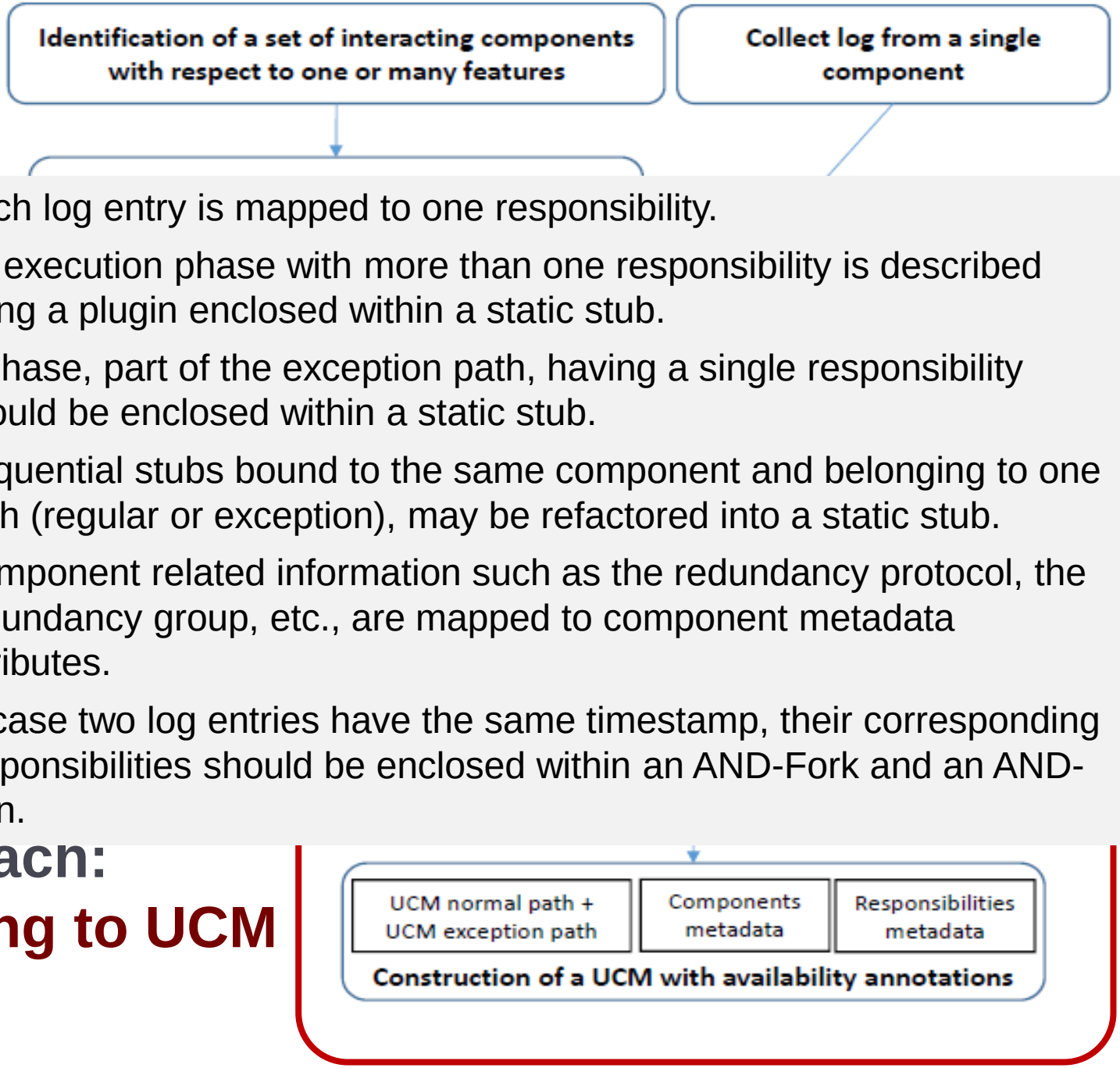


- Log entries from different components are placed in separate phases
- Log entries describing different features' events/errors are placed into separate phases.
- Log entries relative to user actions are separated from system response log.
- Etc.

Approach:
Phase Extraction

Approach:
Mapping to UCM

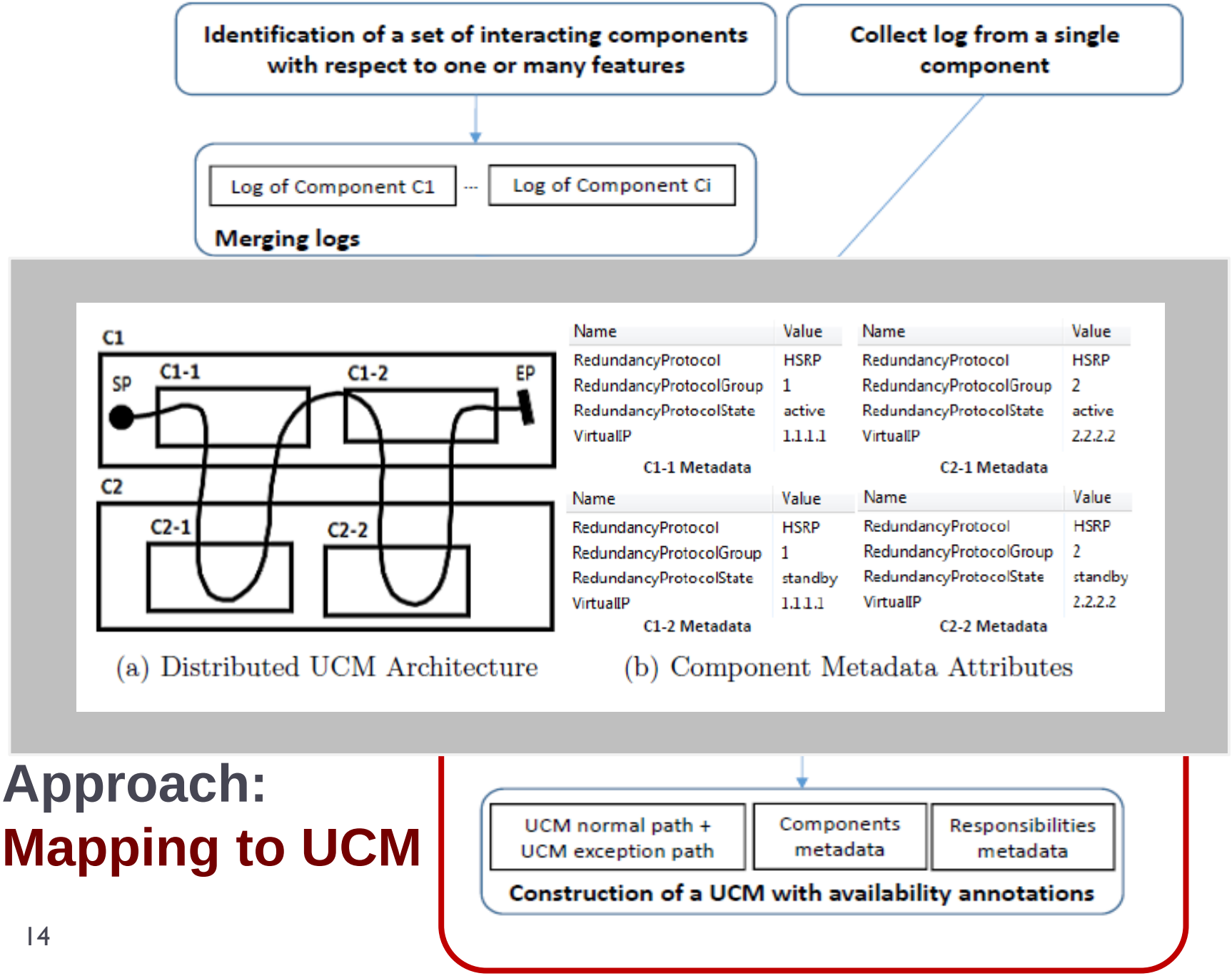




- Each log entry is mapped to one responsibility.
- An execution phase with more than one responsibility is described using a plugin enclosed within a static stub.
- A phase, part of the exception path, having a single responsibility should be enclosed within a static stub.
- Sequential stubs bound to the same component and belonging to one path (regular or exception), may be refactored into a static stub.
- Component related information such as the redundancy protocol, the redundancy group, etc., are mapped to component metadata attributes.
- In case two log entries have the same timestamp, their corresponding responsibilities should be enclosed within an AND-Fork and an AND-Join.

Approacn:
Mapping to UCM

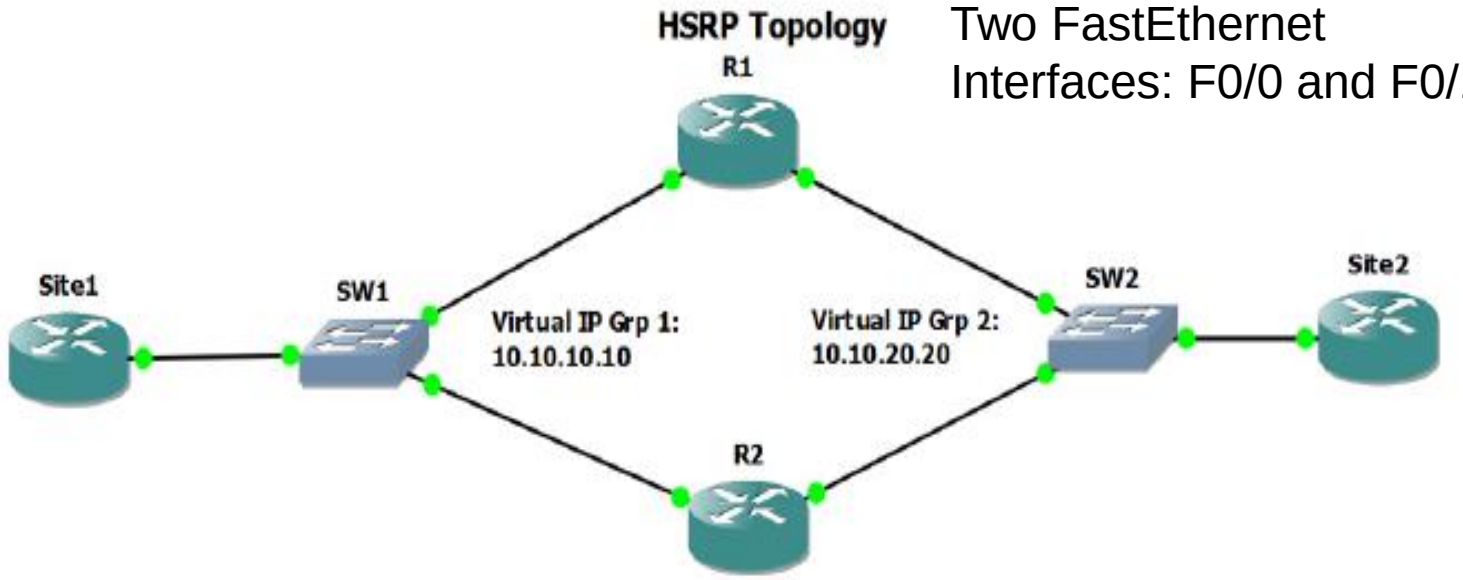
Approach:
Mapping to UCM



Preliminary Evaluation

- Target System: Hot Standby Router Protocol (HSRP)
  - A Cisco proprietary protocol that provides network redundancy for IP networks.
  - By sharing an IP address and a MAC address, two or more routers can act as a single virtual router, known as an HSRP group or a standby group.
  - The active router, elected from the group, is responsible for forwarding the packets that hosts send to the virtual router.
  - If the active router fails, the standby router takes over as the active router. If the standby router fails or becomes the active router, then another router is elected as the standby router.

Testbed: Network topology



Two FastEthernet Interfaces: F0/0 and F0/1

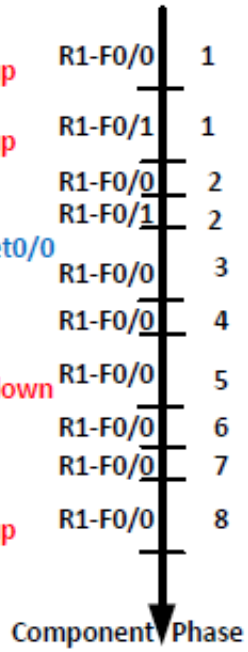
- The testbed was built using the Graphical Network Simulator 3 (GNS3) simulation software. GNS3 allows to emulate complex networks, by combining actual devices and virtual devices together.

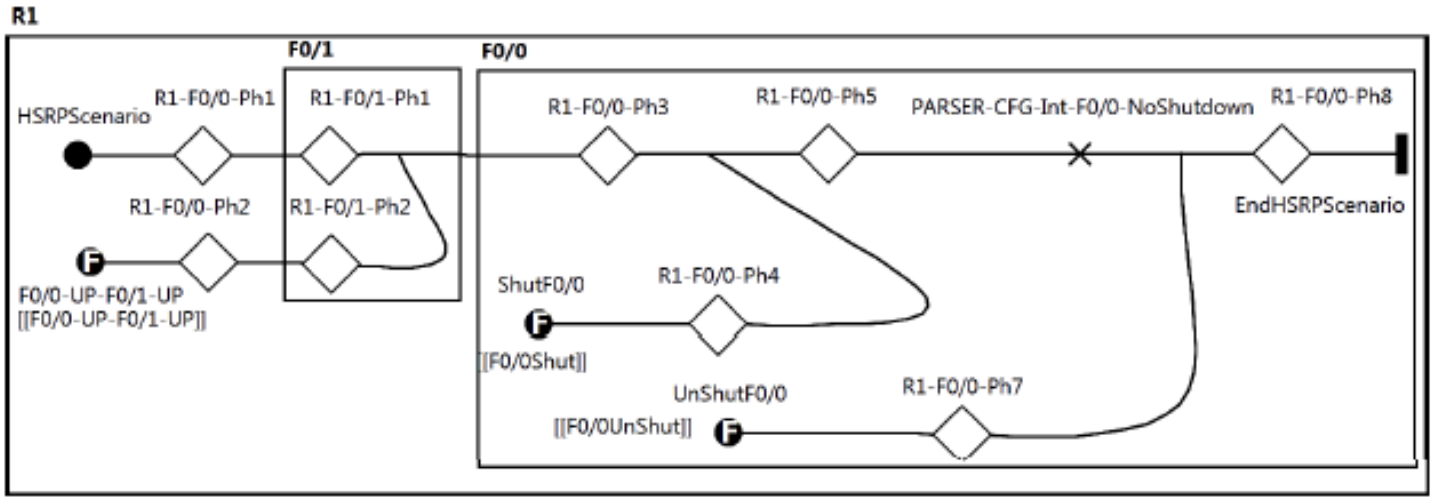
- Logs can be collected from Cisco IOS routers through console logging (default mode), syslog server logging (use of external syslog servers for log storage)…

Sample logs for Router R1

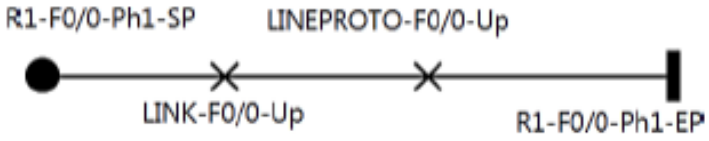
After the log customization and phase extraction steps

*May 27 09:49:51.739: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*May 27 09:49:51.763: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*May 27 09:49:52.863: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*May 27 09:49:52.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
*May 27 09:50:33.063: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active
*May 27 09:50:56.043: %HSRP-5-STATECHANGE: FastEthernet0/1 Grp 2 state Speak -> Standby
*May 27 09:50:57.315: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:interface FastEthernet0/0
*May 27 09:50:58.287: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:shutdown
*May 27 09:50:58.295: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Init
*May 27 09:51:00.267: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
*May 27 09:51:01.267: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
*May 27 09:51:16.447: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:no shutdown
*May 27 09:51:17.931: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Listen -> Active
*May 27 09:51:18.395: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*May 27 09:51:19.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1#

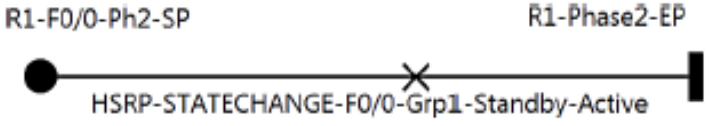




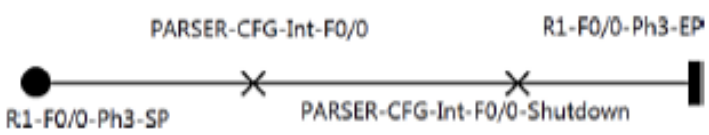
(a) UCM of Router R1



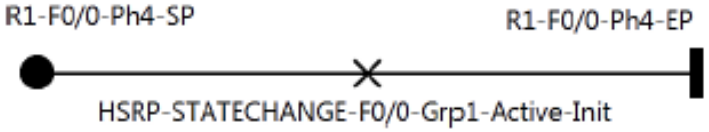
(b) R1-F0/0-Phase1 plugin



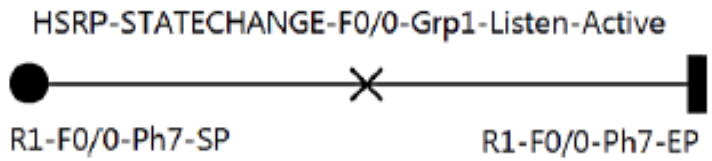
(c) R1-F0/0-Ph2 plugin



(d) R1-F0/0-Ph3 plugin



(e) R1-F0/0-Ph4 plugin



(f) R1-F0/0-Ph7 plugin

Name	Value
AvCat	FaultRecovery
Tactic	Failover

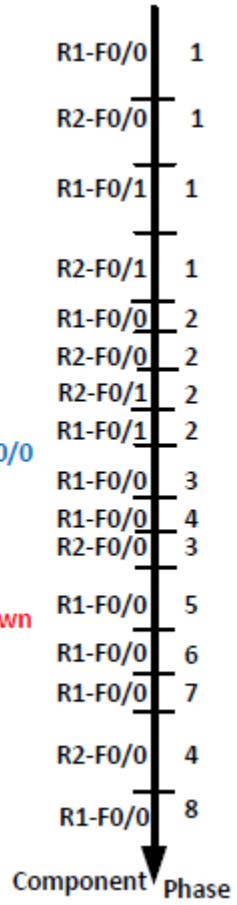
(g) HSRP-STATECHANGE-F0/0-Grp1-Listen-Active metadata

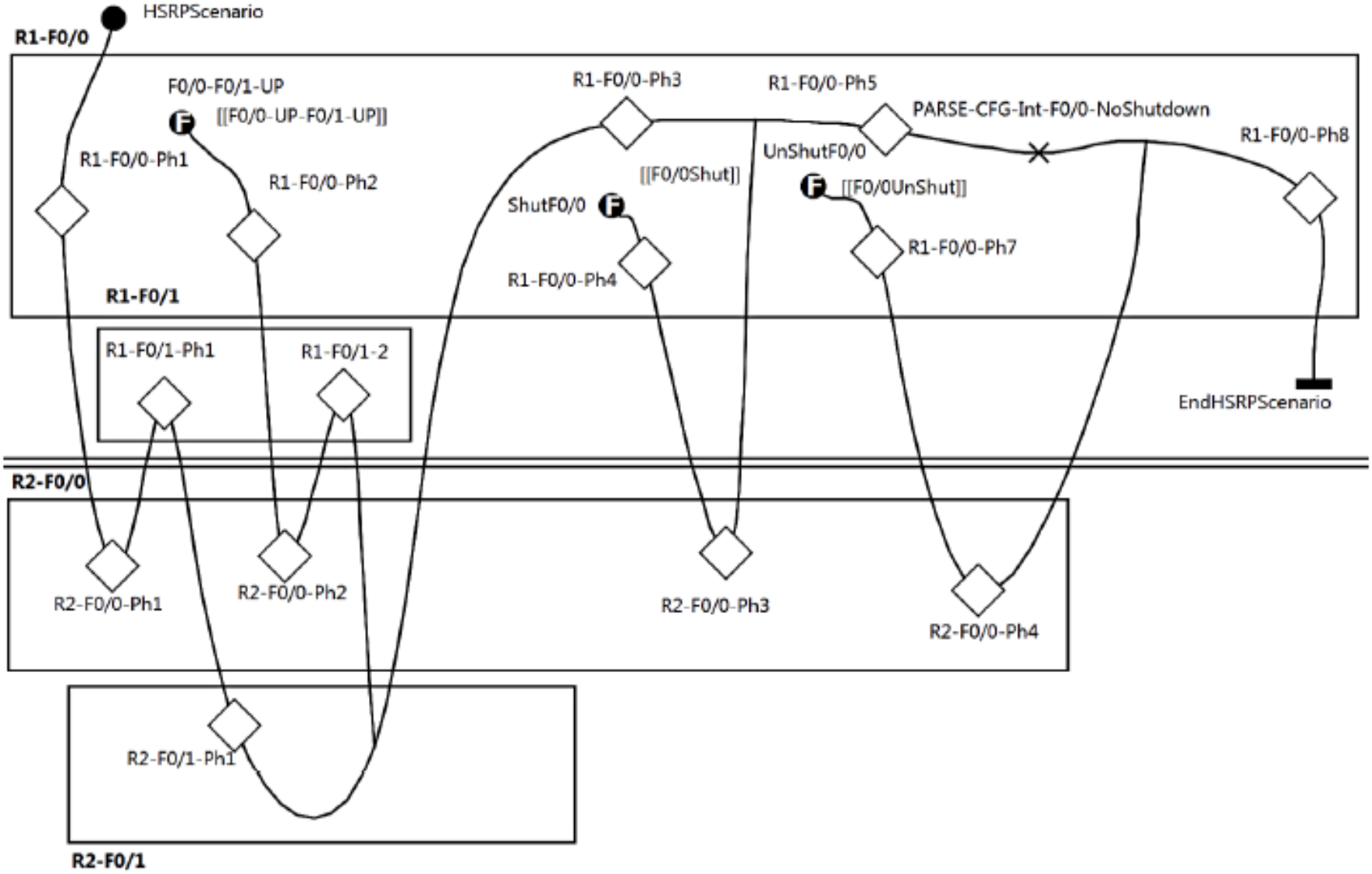
Name	Value
RedundancyProtocol	HSRP
RedundancyProtocolGroup	1
RedundancyProtocolState	active

(h) F0/0 metadata

Combining R1 and R2 Logs

R1*May 27 09:49:51.739: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1*May 27 09:49:51.763: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2*May 27 09:49:52.351: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R2*May 27 09:49:52.371: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1*May 27 09:49:52.863: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
R1*May 27 09:49:52.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R2*May 27 09:49:53.595: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
R2*May 27 09:49:53.603: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1*May 27 09:50:33.063: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active
R2*May 27 09:50:33.979: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Speak -> Standby
R2*May 27 09:50:42.011: %HSRP-5-STATECHANGE: FastEthernet0/1 Grp 2 state Standby -> Active
R1*May 27 09:50:56.043: %HSRP-5-STATECHANGE: FastEthernet0/1 Grp 2 state Speak -> Standby
R1*May 27 09:50:57.315: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:interface FastEthernet0/0
R1*May 27 09:50:58.287: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:shutdown
R1*May 27 09:50:58.295: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Init
R2*May 27 09:50:59.199: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active
R1*May 27 09:51:00.267: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
R1*May 27 09:51:01.267: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
R1*May 27 09:51:16.447: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:no shutdown
R1*May 27 09:51:17.931: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Listen -> Active
R2*May 27 09:51:17.899: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Speak
R2*May 27 09:51:18.395: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Speak -> Standby
R1*May 27 09:51:18.867: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1*May 27 09:51:19.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up




HSRPScenario
R1-F0/0
F0/0-F0/1-UP
[[F0/0-UP-F0/1-UP]]
R1-F0/0-Ph1
R1-F0/0-Ph2
R1-F0/0-Ph3
R1-F0/0-Ph5
PARSE-CFG-Int-F0/0-NoShutdown
R1-F0/0-Ph8
UnShutF0/0
[[F0/0Shut]]
[[F0/0UnShut]]
ShutF0/0
R1-F0/0-Ph7
R1-F0/0-Ph4
R1-F0/1
R1-F0/1-Ph1
R1-F0/1-2
EndHSRPScenario
R2-F0/0
R2-F0/0-Ph1
R2-F0/0-Ph2
R2-F0/0-Ph3
R2-F0/0-Ph4
R2-F0/1-Ph1
R2-F0/1

Conclusion and Future Directions

- We proposed a UCM-based approach for recovering availability requirements from log/traces
  - The approach relies on multiple processing of log information including the extraction of execution phases

- Future Work:
  - **Automation:** investigate how the identification of execution phases can be automated (identify availability patterns, use of heuristics, etc.)
  - **Scalability:** apply the approach to more complex systems (with larger system logs)
  - **Generalization:** generalize the approach to other HA management systems (e.g., AMF from SAForum)
  - **Usability:** work with analysts to assess the usability and utility of the approach in practice

Thank you.