

Challenges in Dependability Evaluation of Large Scale Systems

Karama Kanoun



IFIP WG 10.4, 54th meeting, Alyeska, Alaska — June 27-30, 2008

Context

☞ Objectives of (benefits from) dependability evaluation

- ◆ B1: Select appropriate system / architecture / recovery techniques / tradeoffs
- ◆ B2: Characterize system dependability in a rational way
- ◆ B3: Plan maintenance
- ◆ B4: Improve system dependability
- ◆ B5: Formulate service level agreements
- ◆ B6: On-line monitoring

☞ Approaches

- ◆ Measurements, experimentation, modeling

☞ Profitable to all actors ⇒ consistent views ⇒ **integrated approaches**

- ◆ System users, designers, providers, regulators

Current State of Practice

☞ Safety critical systems, control systems, small and medium size

- ◆ Common use

- ◆ B1 - B4

- ◆ Analyses performed during system design or operation

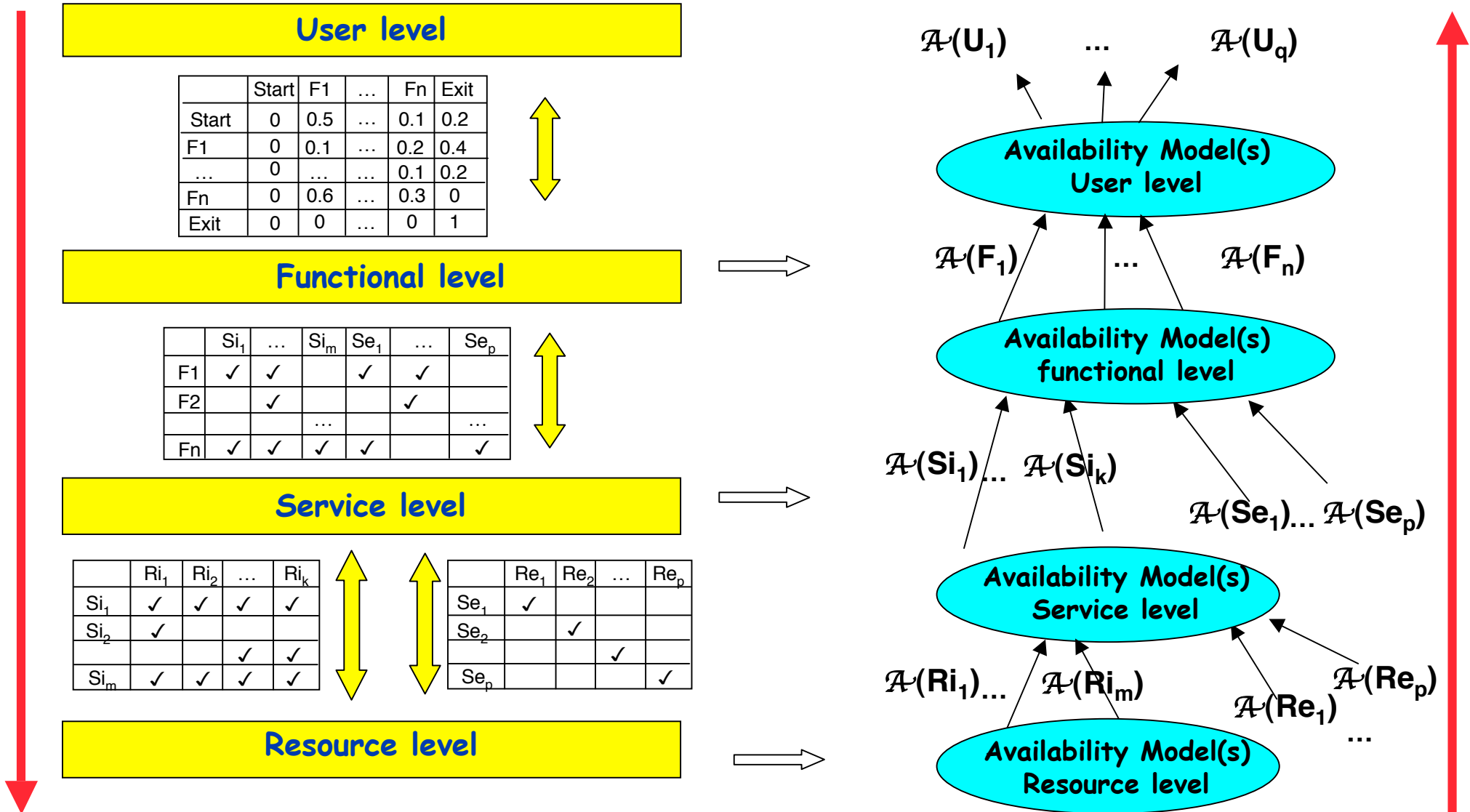
- ◆ After updates, new analyses?

☞ Large scale systems (LSS)

- ◆ Classical approaches: compositional, hierarchical

- ◆ End-to-end scenarios

Hierarchical Approaches for Modeling LSS Dependability



Hierarchical Approaches for Modeling LSS Dependability

User level

	Start	F1	...	Fn	Exit
Start	0	0.5	...	0.1	0.2
F1	0	0.1	...	0.2	0.4
...	0	0.1	0.2
Fn	0	0.6	...	0.3	0
Exit	0	0	...	0	1

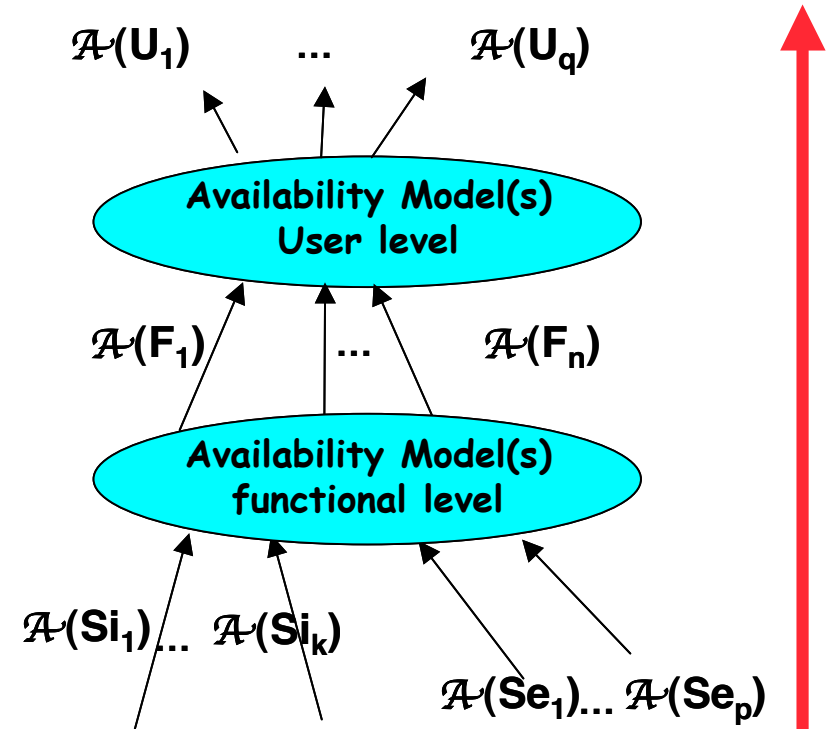
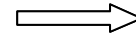
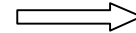


Functional level

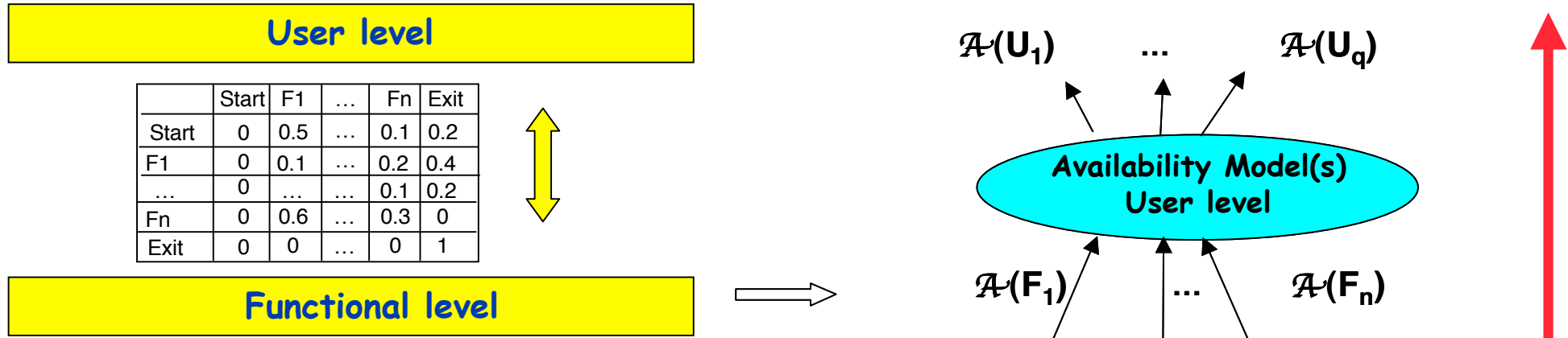
	Si ₁	...	Si _m	Se ₁	...	Se _p
F1	✓	✓		✓	✓	
F2		✓			✓	
...		
Fn	✓	✓	✓	✓		✓



Service level



Hierarchical Approaches for Modeling LSS Dependability



Difficulties when Modeling LSS Dependability

- ☞ System structure and boundaries unknown
- ☞ Continually evolving systems
- ☞ Large number of users with different (unknown?)
 - ◆ Expectations
 - ◆ Profiles
 - ◆ Skills
 - ◆ Mobility
- ☞ Service(s)?
- ☞ Measure(s) of dependability?
- ☞ Prediction?

Challenges in Modeling LSS Dependability

- ☞ High level abstraction modeling
- ☞ Assessment in the presence of perpetual evolution
 - ◆ Short-term, Medium-term, Long-term
- ☞ On-line evaluation as a support for
 - ◆ Diagnosis
 - ◆ Selection of system (re)configuration
 - ◆ Adaptation to changes / evolutions
- ☞ Security evaluation
- ☞ Dependability benchmarks?