FORMAL METHODS FOR SYSTEM VERIFICATION

Introduction to the course

Sabina Rossi

DAIS
Università Ca' Foscari
Venezia

Key notions

- A model can be constructed to represent some aspects of the dynamic behaviour of a system.
- Once constructed, such a model becomes a tool with which we can investigate the behaviour of the system.

Descrete event systems

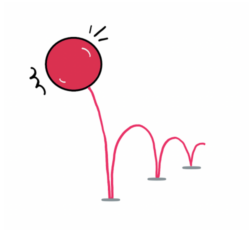
- In this course we will consider discrete event systems.
- What is a discrete event system ?
- A system whose state changes based upon the occurrence of discrete events.

Time driven vs. Event driven

Time driven: Bouncing ball

- State of the system is dependent on time and physical parameters:
    - initial height of ball
    - initial velocity of ball
    - gravity

This is actually a deterministic system.

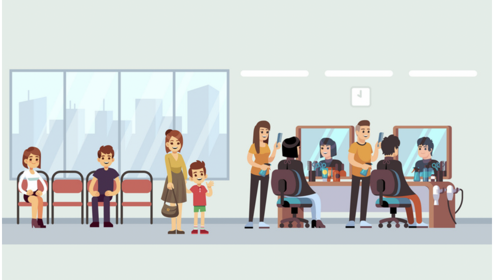


Time driven vs. Event driven

Event driven: Barber shop

- This is an event driven system in which the state of the system does not change unless an event occurs:
  - customer arrives
  - customer queues
  - customer leaves

  This is actually a non-deterministic system.



Descrete event systems

Characterising a descrete event systems

- The state of the system is characterised by variables which take distinct values and which change by discrete events, i.e. at a distinct time something happens within the system which results in a change in one or more of the state variables.

Example

- We might be interested in the number of nodes N in a communication network which are currently waiting to send a message
  - If a node, which was not previously waiting, generates a message and is now waiting to send then $N \rightarrow N + 1$.
  - If a node, which was previously waiting, successfully transmits its message then $N \rightarrow N - 1$.

Descrete event systems

Discrete time vs. Continuous time

- Within discrete event systems there is a distinction between a discrete time representation and a continuous time representation:

  Discrete time: such models only consider the system at predetermined moments in time, which are typically evenly spaced, e.g., at each clock "tick".

  Continuous time: such models consider the system at the time of each event so the time parameter in such models is conceptually continuous.

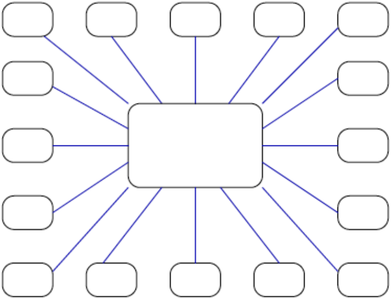
Continuous time models are generally appropriate for computer and communication systems.

Performance modelling

Modelling computer systems

- Performance modelling is concerned with the description, analysis and optimisation of the dynamic behaviour of computer and communication systems.
- The aim is to understand the behaviour of the system and identify the aspects of the system which are sensitive from a performance point of view.

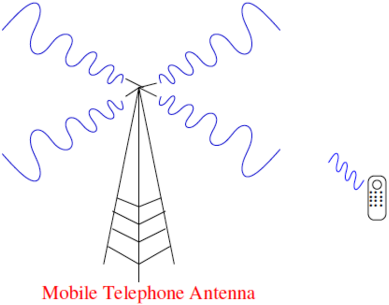
Performance Modelling: Motivation



Capacity planning

- How many clients can the existing server support and maintain reasonable response times?

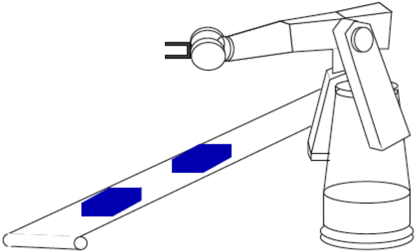
Performance Modelling: Motivation



System Configuration

- How many frequencies do you need to keep blocking probabilities low?

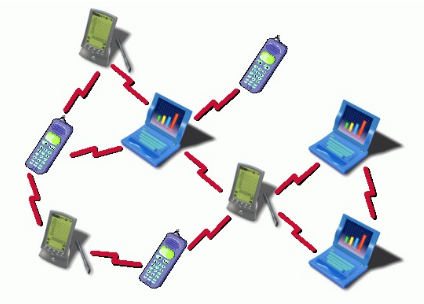
Performance Modelling: Motivation



System Tuning

- What speed of conveyor belt will minimize robot idle time and maximize throughput whilst avoiding lost widgets?

Performance Modelling: Motivation



Sustainable energy planning

- Which is the minimum transmission radius to ensure connectivity while optimizing energy consumption?

Performance Modelling: Motivation

Prediction

- Which is the expected transaction consolidation time with respect to the offered fee?

Performance Modelling

- Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and efficient sharing of resources.
- This often involves a trade-off between the interests of the users, who want more resources and the interests of system operators, who want to minimize the resources.

Performance Modelling

There are often conflicting interests at play:

- Users typically want to optimise external measurements of the dynamics such us
  - response time (as small as possible)
  - throughput (as high as possible)
  - blocking probability (preferably zero)
- In contrast system managers may seek to optimize internal measurements of the dynamics such as
  - utilisation (reasonably high, but not too high)
  - idle time (as small as possible)
  - failure rates (as low as possible).

Performance Modelling

- It has been applied to computer systems since the mid-1960s and communication systems since the early 20th century.
- Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed.
- But as computer systems have developed these techniques are no longer widely applicable for expressing the dynamic behaviour observed in distributed systems.

Does timeliness matter...?

- There is sometimes a perception in software development that performance does not matter much, or that it is easily fixed later by buying a faster machine.
- On the contrary — studies have shown that response time is a key feature in user satisfaction and trust in systems.
- In a recent study by Amazon they artificially delayed page loading times in increments of 100 milliseconds. Even such very small delays were observed to result in substantial and costly drops in revenue.
- Google, Ebay, Vodafone, report similar findings.

See:
`https://www.conductor.com/academy/page-speed-resources/`

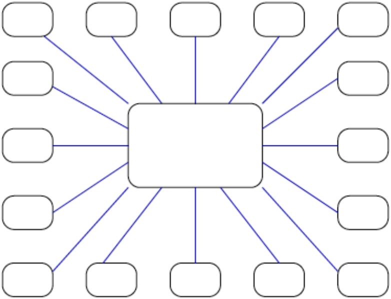
Modelling computer systems: the challenges

- Physical distance – need to represent time
  - Network latency
- Partial failures – randomness and probability
  - Server may be down
  - Routers may be down
- Scale – need to quantify population sizes
  - Workload characterisation
- Resource sharing – need to express percentages
  - Network contention
  - CPU load

Modelling computer systems: the challenges

Time	What representation of time will we use?
Randomness	What kind of random number distributions will we use?
Probability	How can we have probabilities in the model without uncertainty in the results?
Scale	How can we escape the state-space explosion problem?
Percentages	What can it mean to have a fraction of a process?

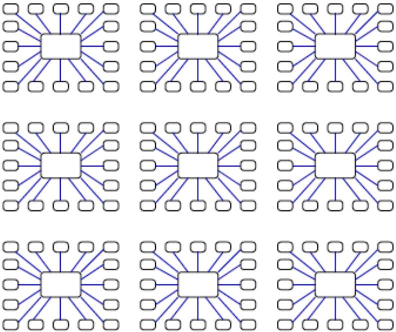
Quantitative Modelling: Motivation



Quality of Service issues

- Can the server maintain reasonable response times?

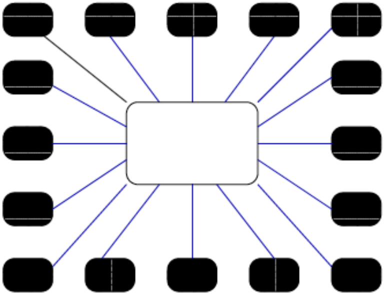
Quantitative Modelling: Motivation



Scalability issues

- How many times do we have to replicate this service to support all of the subscribers?

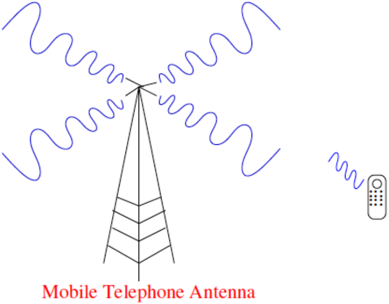
Quantitative Modelling: Motivation



Robustness and scalability issues

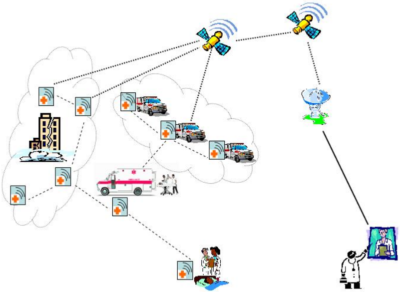
- Will the server withstand a distributed denial of service attack?

Quantitative Modelling: Motivation



Service-level agreements

- What percentage of downloads do complete within the time we advertised?



Scalability issues

- Which is the optimal router allocation?

Performance Modelling: Motivation

Trade-off between costs and response times

- Which is the relation between the fee offered by a transaction and its expected consolidation time?

Quantitative Modelling

- When systems are modelled to verify their functional behaviour (correctness), all definite values are abstracted away $\rightarrow$ qualitative modelling.
- In contrast, performance modelling is quantitative modelling as we must take into account explicit values for time (latency, service time etc.) and probability (choices, alternative outcomes, mixed workload).
- Probability will be used to represent randomness (e.g., from human users) but also as an abstraction over unknown values (e.g., service times).

Performance modelling

- In performance modelling an abstract representation, or model, of the system is used to capture the essential characteristics of the system so that its performance can be reproduced.
- Typical models are queueing networks and stochastic Petri nets which are both based on stochastic models.
- In many cases, the underlying stochastic models are assumed to be Markov processes.

Process Algebras

- Process algebras are mathematical theories which model concurrent systems by their algebra and provide apparatus for reasoning about the structure and behaviour of the model.

- Process algebras where models are decorated with quantitative information used to generate a stochastic process are called stochastic process algebras (SPA).

Process Algebras

- Examples of process algebras include:
  - CCS - Calculus of Communicating Systems
  - CSP - Communicating Sequential Processes
  - ACP - Algebra of Communicating Processes
  - PEPA - Performance Evaluation Process Algebra

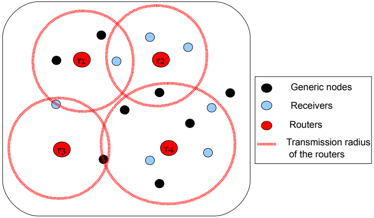
- A system is characterised by its active components and the interactions, or communications, between them.

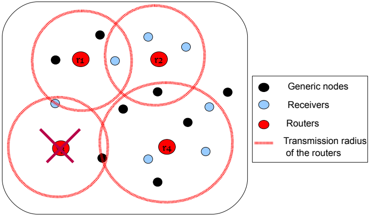
Pure process algebras $\rightarrow$ qualitative modelling

- The models are used to establish the correct behaviour of systems, both with respect to a given specification and in the more abstract sense.
- Time is abstracted away within a process so that all actions are assumed to be instantaneous and only relative timing is represented via the traces of the process.
- Behavioural properties such as fairness and freedom from deadlock are investigated, in contrast to the quantitative values extracted from performance models.

Connectivity properties

The following two networks are equivalent in the sense that they exhibit the same connectivity properties wrt the receivers





Probabilistic/Stochastic Process Algebras

- Process algebras will often be used to model systems in which there is uncertainty about the behaviour of a component but, like time, this uncertainty will be abstracted away so that all choices become nondeterministic.
- Probabilistic extensions of process algebras allow this uncertainty to be quantified because nondeterministic choice is replaced by a probabilistic choice.
- In this case a probability is associated with each possible outcome of a choice.

Process Algebra for Performance Modelling

Motivations

Integrating Performance Analysis into System Design: It is important to consider the timely consideration of performance aspects of a planned system.

Representing Systems as Models: The restricted expressiveness of queueing networks has been highlighted by recent developments in computer and telecommunication systems.

Model Tractability: Solving models of the size and complexity necessary to model many modern systems is often beyond the capabilities of contemporary techniques and equipment. This has led to considerable interest in model simplification and aggregation techniques.

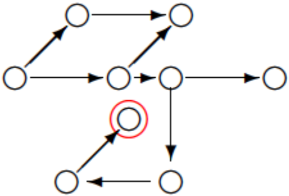
Process Algebras as a Design Methodology

- The process algebra style of system description is close to the way that designers think about systems.
- Using a process algebra based language for performance modelling introduces the possibility of a closer integration of performance analysis into design methodologies.
- It allows to perform both qualitative (or functional) and quantitative modelling using the same system description.

Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

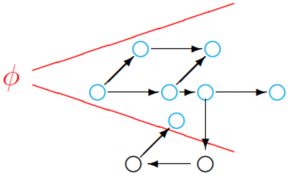
Reachability analysis



- How long will it take for the system to arrive in a particular state?

Qualitative verification can now be complemented by quantitative verification.

Model Checking

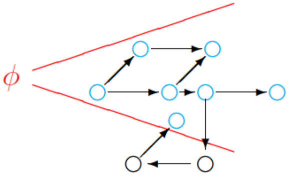


- Does a given property ϕ hold within the system with a given probability?

Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Model Checking



- For a given starting state how long is it until a given property ϕ holds?

The "Cooperator" Paradigm

- A process algebra description represents a system as a collection of active agents who cooperate or interact to achieve the behaviour of the system.
- For example, in distributed systems and communication networks components have autonomy and the framework is one of cooperation.
- In a process algebra model all system elements have equal status; the model defines their individual behaviours and how they interact.

Hierarchical Models

- Process algebras include mechanisms for composition and abstraction, as well as apparatus for compositional reasoning, which are missing from performance modelling techniques.
- These mechanisms, which are an integral part of the language, facilitate the systematic development of large models with hierarchical structure.

The PEPA project

The PEPA project started in Edinburgh in 1991.

Motivation: the size and complexity of many modern systems result in large, complex models.

Idea: A compositional approach decomposes the system into subsystems that are smaller and more easily modelled.

Objective: The PEPA project aims at providing a novel compositional approach to performance modelling.

Result: PEPA is a stochastic process algebra.

References

- PEPA:

  http://www.dcs.ed.ac.uk/pepa/

- Textbook: J. Hillston. A Compositional Approach to Performance Modelling. Cambridge University Press, 1996.

  http://www.dcs.ed.ac.uk/pepa/papers/