IKR SimLib –
The IKR Simulation Library

An Overview

mail@ikr.uni-stuttgart.de
August 2008

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Dr. h.c. mult. P. J. Kühn
Editors

Stefan Bodamer  Klaus Dolzer  Christoph Gauger
Marc Necker  Jörg Sommer  Joachim Scharf
Overview

- motivation
- classification of evaluation methods
- simulation basics
  - event-driven simulation
  - random number generation
  - statistical evaluation
- IKR simulation library
  - overview on the simulation environment
  - modelling concepts
  - measurement and evaluation
  - simulation control
  - parallel simulation
- practical usage
  - integration into development cycle
  - performing simulation studies
- extensions of the IKR simulation library
- sample problems solved using the IKR simulation library
Motivation

Communication Network Design Issues

requirements
• performance
  – response time
  – loss rate
  – throughput
  – fairness
  – signal quality
• availability
• security

influence on
• system dimensioning
• system structure
• protocols
Motivation

Simulation in Other Areas

examples

• computer systems evaluation
• analysis of production systems
• dimensioning and operation of transport systems (road network, airport, railway system, seaport)
• evaluation of operational organisation in hospitals, public offices, self-service restaurants, ...
• analysis of financial systems
• astronomy/meteorology
• fluid mechanics
• process sequence in a power plant
• aging process of system components
Classification of Evaluation Methods

Overview

- system
  - evaluation of a real system
    - physical model
  - evaluation of a model
    - formal model
      - analytical methods
      - simulation

according to [1.]
Classification of Evaluation Methods

Real System – Model

Evaluation of a real system

- Requirements
  - Existence of the system
  - Reproducibility
  - Accessibility (space/time)
  - No danger
  - Low cost
- Examples
  - Traffic measurements (telecommunication, road traffic)
  - Evaluations with prototypes
  - Tests of software products

Evaluation of a model

- Requirements
  - Knowledge about the system (which parts are relevant?)
  - Validity of the model
  - Illustration
  - Tractability
- Characteristic measure
  - Grade of abstraction
Classification of Evaluation Methods

Physical Model – Formal Model

Physical model
• physical imitation of the system
  – as a whole or partly
  – omitting irrelevant parts
  – simplification
• examples
  – automotive development (wind tunnel, crash test)
  – virtual cockpits
  – biological evaluations
  – sociology
  replay of situations by persons

Formal model
• functional description
  – SDL
  – state transition diagrams
  – Petri nets
• queueing model
  stochastic arrival process
  queue
  service strategy
  stochastic service process
Classification of Evaluation Methods

**Analytical Methods – Simulation**

**analytical methods**
- exact or approximate solution of mathematical terms
- typical
  - systems of linear equations
  - differential equation systems
  - eigenvalue and boundary value problems
- critical
  - existence and robustness of a solution algorithm
  - complexity and accuracy of numerical evaluation

**simulation**
- replaying relevant procedures on a computer
- typical
  - high number of events
  - usage of random numbers
  - measurement of samples
  - statistical evaluation
- critical
  - statistical robustness
  - transient phases
  - memory/computation time
  - implementation faults
Classification of Evaluation Methods

Classification Criteria

- statistical – functional
- dynamic – static
- discrete – continuous
- event-driven – time-driven
- stochastic – deterministic
- stationary – instationary
- sequential – parallel
- in software – in hardware
Simulation Basics

Event-Driven Simulation

- system state changes at discrete points in time (e.g. arrival of a message, end of service, timer expiration)
- state change represented by an event
- registration of events in a calendar (together with a timestamp representing planned time of occurrence)
- sequential processing of events in the calendar by order of timestamps
  - update simulated system time
  - execute actions to realize state changes
  - plan successive events

© 2008 Universität Stuttgart • IKR
IKR SimLib – An Overview
Simulation Basics

Example: M/M/1 Single Server System

- next event: arrival of a request
- new system time: 17.12
- state change
  - occupy server (if idle) or enter queue
  - update of statistics
- successive events
  - next arrival event + (possible) end of service
  - timestamp: current system time + random duration according to corresponding distribution
  - register in the calendar
Simulation Basics

Generation of Random Numbers

- usage of random numbers, e.g. for
  - interarrival time
  - service time
  - batch size of batch arrivals
- classification according to nature of random numbers
  - continuous random variables (e.g. negative-exponential distribution)
  - discrete random variables (e.g. Poisson distribution, geometric distribution)
- generation of a (0,1)-uniform distributed random number
  - pseudo random number generator: deterministic algorithm for computing a random number with given starting value
  - example: linear congruential random number generator with sequence
    \[ Z_i = (a \cdot Z_{i-1} + c) \mod m \]
    yields random no. \( U_i = Z_i / m \) (e.g. \( m = 2^{31} - 1 \))
  - advantages: reproducibility, simple generation
  - disadvantages: periodicity, autocorrelation
Simulation Basics

Generation of Random Numbers (2)

- generation of a random number according to a distribution function
  - inversion of the distribution function \( F(x) = P\{X \leq x\} \)

  \[
  F(x) = \begin{cases} \lambda & \text{if } x = 0 \\ \frac{1}{\lambda} \cdot \ln(1-U) & \text{if } x > 0 \end{cases}
  \]

- example: random number for negative-exponential distribution function
  \( F(x) = 1 - e^{-\lambda x} : X = \frac{1}{\lambda} \cdot \ln(1-U) = \frac{1}{\lambda} \cdot \ln U' \)

- alternatives
  - generator method: generation of a random number from several random numbers with invertible distribution (example: Erlang-k-distribution)
  - approximation (e.g. normal distribution)

- generation of values for correlated random variables according to a source model (e.g. On-Off, MMPP)
Simulation Basics

Statistical Evaluation

- examples for measured data
  - transfer time
  - queue length
  - loss

- evaluation according to
  - moments (mean value, variance, coefficient of variation)
  - relative occurrence (distribution and density)
  - quantile (e.g. 99% quantile \( x_{0.99} \): \( F(x_{0.99}) = P\{X \leq x_{0.99}\} = 0.99 \) )
  - coefficient of (auto-)correlation/(auto-)covariance

- realization according to estimation functions or algorithms
Estimation of Statistical Measures

- mean: $\bar{X} = \frac{1}{m} \cdot \sum_{i=1}^{m} X_i$ ($X_i$: measured samples, $m$: number of samples)

- variance: $S^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} (X_i - \bar{X})^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} X_i^2 - \frac{1}{m} \cdot (m-1) \cdot \sum_{i=1}^{m} X_i$

- distribution (histogram)
  - divide relevant range of values $[1, u]$ into $k$ intervals with size $w = \frac{u-1}{k}$
  - count number of samples falling into each interval: $m_i$ ($i = 1, 2, \ldots, k$)
  - count number of samples smaller than $1$: $m_1$
  - estimator for $P\{X \leq 1 + j \cdot w\}$:
    $\frac{1}{m} \cdot \left( m_1 + \sum_{i=1}^{j} m_i \right)$ for $j = 0, 1, \ldots, k$
Simulation Basics

Statistical Significance

• questions
  – how exact are the simulation results compared to the real values?
  – how long does a simulation have to last in order to reach sufficient exactness?
• commonly used: batch means method
  – partitioning of simulation into $n$ batches (e.g. $n = 10$)
  – assumption of statistical independence between batches
  – interpretation of batch results as samples
  – averaging and computation of the confidence interval

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} \bar{X}_i$$

queue length $X(t)$

transient phase

1st batch

2nd batch

... 

nth batch

$X_1$ 

$X_2$ 

$X_n$

time
Simulation Basics

Computation of Confidence Intervals

• related to confidence level $\alpha$ (e.g. $\alpha = 0.95$: 95% confidence interval)
• interpretation: actual value $\mu$ is located within the confidence interval $[\bar{X} - \varepsilon, \bar{X} + \varepsilon]$ with probability $\alpha$, $\rightarrow P\{\bar{X} - \varepsilon \leq \mu \leq \bar{X} + \varepsilon\} = \alpha$
• assumptions for batch results $\bar{X}_i$ (valid if batches are long):
  • quasi-independent
  • quasi-normal distribution (due to central limit theorem)
• random variable $T = (\bar{X} - \mu) / \sqrt{S^2/n}$ is student-t-distributed with order $n - 1$
  $(n = \text{no. of batches, } S^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (\bar{X}_i - \bar{X})^2 = \text{empirical variance})$
• $\varepsilon$ and thus the confidence interval can be obtained from
  $$\varepsilon = t_{n-1}(\frac{1 + \alpha}{2}) \cdot \sqrt{\frac{S^2}{n}}$$
  where $t_{n-1}(\beta)$ denotes $\beta$ quantile of the student-t-distribution of order $n - 1$
The IKR Simulation Library

Overview

• history
  – origin: Pascal simulation library (1980ies)
  – since then enhanced and improved continuously

• implementation
  – C++ class library (for class hierarchy see [2.])
  – Java library (for class hierarchy see [2.])
  – usage of additional libraries (e.g., container class library)
  – currently available for Linux, but almost no platform dependent code

• main features
  – support for transformation of an abstract model into source code
  – control of event-driven simulation
  – random number generation (various distributions and source models)
  – statistical evaluation
  – reading parameter values and printing results

• Responsible persons for support, maintenance, and development
The IKR Simulation Library

Basic Structure

user's simulation model

standard components
  model components
  utilities

modeling concepts
  basic entity
  port concept
  filters & meters

basic concepts
  simulation control
  event handling
  random distributions
  statistical evaluation
  I/O concept
The IKR Simulation Library

Model Components: Class Hierarchy

- PrintServer
- Entity
  - Generator
    - StdGenerator
    - Stream Generator
  - Queue
    - FIFOQueue
    - UnboundedFIFO Queue
    - BoundedFIFO Queue
  - Multiplexer
    - Static Multiplexer
    - Dynamic Multiplexer
The IKR Simulation Library

Model Components: Object Hierarchy

decomposition support by
- hierarchy
  - has-relationship
  - pointer to owner
- name concept
  - local name as attribute
  - identification of components
  - access via central component manager
The IKR Simulation Library

Modeling Components: Port Concept

- message exchange between model components via ports
  - distinction between input and output ports
  - central port registration using owner address and port name
  - connection of the ports using function call `Connect`
  - communication via handshake protocol

→ uniform interface for model components
The IKR Simulation Library

Exchanging Messages: Example

sender

handleGetMessage()

msg

messageIndication()

queue

input: InputPort

output: OutputPort

server

handleMessage()

msg

msg

messageIndication()

msg

msg

get Message()

msg

[ free ] getMessage()

[ answer ] getMessage()

get Message()

isMessageAvailable()

isMessageAvailable()

isMessageAvailable()

endOfService()

active

receiver

handleGetMessage()

msg

handle

answer

handle

answer

handle

answer

nextMsg()
The IKR Simulation Library

Measurements: Message Filters

- attached to port using a method call
- redirection of protocol functions

- motivation
  - evaluation or modification of message contents
  - message tracing
  - applied in meters
The IKR Simulation Library

Measurements: Meters

- attached to ports via message filters
- application
  - message counting/rate measurement
  - transfer time measurement

- internal evaluation of measured values using statistic classes
The IKR Simulation Library

Statistical Evaluation

- various statistic classes
  - sample moment (mean, variance)
  - time-based mean
  - quantile
  - probability
  - distribution (histogram)
  - correlation
- computation of confidence intervals
- application in
  - meters (e.g. transfer time)
  - model components (e.g. queue length)
- connection to simulation control
**Simulation Control**

- execution of simulation by calling method `Run` in the simulation object
- initiation
- transient phase
- batches
- preparation and output of results
- access to simulation control via notifier
  - determine end of batch/simulation
  - based on message count or time
- triggering objects at beginning and end of phases, e.g.
  - statistics
  - generators
The IKR Simulation Library

Sequential Simulation

\[ t_{seq} = t_{\text{transient}} + n \cdot t_{\text{batch}} \]

Parallel Simulation

- parallel simulation in case of
  - simulation points with large simulation time
  - enough computing capacity
- independent simulation runs on different CPUs and/or cores
  \[ t_{\text{parallel}} = t_{\text{transient}} + k \cdot t_{\text{batch}} \]
  with \( 1 \leq k \leq n \)
- disadvantage: overhead due to multiple transient phases
Parallel Simulation: Realization

1st. Step

Simulation program

command line

par file

batch export file

batch export file

batch export file

2nd. Step

Simulation program

command line

par file

fmt file

final log file
Practical Usage

Integration into Development Cycle

- system description
- traffic/queueing modelling
- OOA/OOD
- implementation
- validation
- simulations
- evaluation visualization
Practical Usage

Performing Simulation Studies

- problem specific code
- simulation library
- add-on libraries
- input parameters
- simulation program
- results
- visualization
Practical Usage

Simulation Programs

- problem-specific model and design
- typically several thousand lines of code

C++ Edition

Simulation Program
- TCP Library
- Emulation Library
- Simulation Library
- Component Library
- Utility Library

libraries with more than 400 classes and more than 70,000 lines of codes

Java Edition

Simulation Program
- Emulation Library
- Simulation Library
- Java's Base Libraries (e.g. Collections)

libraries with more than 400 classes and more than 70,000 lines of codes
TCP Library: Overview

- models unidirectional TCP data communication
- all important protocol mechanisms
  - reliable data delivery
  - flow control and congestion control
  - different TCP algorithms (Reno, NewReno, Tahoe, ...)
- light-weight implementation
  - simple connection setup and release
  - some enhancements are omitted, e.g., selective acknowledgments and explicit congestion notification
- flexible integration of application models
- results comparable to other simulation environments, such as ns-2 (UC Berkeley, LBL, USC/ISI, and Xerox PARC)
- currently, only available in C++ (and not in Java)
TCP Library: Basic TCP Simulation Setup

- packet loss
- delay
- delay jitter
- packet reordering
Extensions of the Simulation Library

Emulation Library

- integrated simulation and emulation environment
  - use simulation to optimize system over a large parameter space
  - use emulation to evaluate the performance of real world applications
- reuse of existing simulation tools
- example: WWW access over GPRS and UMTS

![Diagram](image_url)
Simlib Usage in Projects

Overview

• optical networks
  – comparison of reservation concepts for optical burst switching
  – evaluation of dynamic routing strategies in WDM networks
• mobile communication networks
  – performance of packet traffic in radio access networks (GPRS, UMTS, ...)
  – capacity planning and network dimensioning
• Ethernet networks
  – dimensioning of switch buffers
  – evaluation of standard extensions (scheduling algorithms, frame extensions, ...)
• IP networks
  – IP networks with service differentiation
  – comparison of performance measures for TCP-based IP traffic
• ATM networks
  – dimensioning of ATM switches
  – investigation of traffic management mechanisms (CAC, ABR, GFR, ...)
• in-vehicle communication networks
Simlib Usage in Projects

Example 1: ATM Switch

- **system**
  - **modelling:** queueing network
  - stochastic arrival process
  - queue
  - multiplexer
  - stochastic service process

- **issues**
  - cell losses
  - delay/jitter
  - influence of the bus protocol
  - buffer dimensioning
Simlib Usage in Projects

Example 2: Switching Software

- system

- modelling: state machine

- issues
  - connection setup time
  - load caused by software components
  - overhead of operating system
  - influence of component distribution
Simlib Usage in Projects

Example 3: UMTS Performance

- system

- modelling: multi-layer model

- issues
  - system dimensioning, cross-layer protocol optimization
  - transport and application layer performance

---

**UTRAN**

- **RNC**: Radio Network Controller
- **SGSN**: Serving GPRS Support Node
- **GGSN**: Gateway GPRS Support Node

**Core Network**

- **Internet**
- **Server**

**Data Flow**

- **TCP/IP**
- **PDCP**
- **RLC**
- **MAC**
- **PHY**

**Network RTT**

- long RTT: >120ms
- short – medium RTT: ~10–200ms
Simlib Usage in Projects

Example 3: Simulation/Emulation Model

- single cell
- single user
- ideal power control
Simlib Usage in Projects

Example 3: Scenario with Proxy

Diagram showing a network scenario with Proxy, including UTRAN, Core Network, Internet, Server, and various nodes such as Node B, RNC, SGSN, GGSN, Proxy, Mozilla Web Browser, and Apache Web Server.
Simlib Usage in Projects

**Example 3: Some Simulation Results**

- FTP download
- TCP NewReno
- Downlink
  - DCH 256 kBit/s
  - 20% blocking probability
- Uplink
  - DCH 64 kBit/s
  - 10% blocking probability

- large delays in UMTS slows down recovery from Internet packet loss
- TCP proxy decouples radio access network from Internet
Practical Usage

Example 4: Ethernet networks

- system

- issues
  - system dimensioning (e.g. buffer size, link capacities)
  - latency and jitter (temporal requirements)
  - evaluation of different schedulers
  - frame loss
Further Reading

2. IKR Simulation Library: http://www.ikr.uni-stuttgart.de/IKRSimLib.