Compositional modelling of concurrent systems and their quantitative evaluation

Rocco De Nicola

Università degli Studi di Firenze
visiting IMT - Lucca

Camerino
September 15, 2010

based on joint work with:
M. Bernardo, D. Latella, M. Loreti, M. Massink
Outline...

1. Introduction and Motivation
2. Rate Based Transition Systems
3. An RTS Semantics for Pepa
4. ULTRAS
5. Conclusions
To deal with reactive systems and guarantee their correct behavior in all possible environment, we need:

1. To study **mathematical models** for the formal description and analysis of concurrent programs.
To deal with reactive systems and guarantee their correct behavior in all possible environment, we need:

1. To study mathematical models for the formal description and analysis of concurrent programs.

2. To devise formal languages for the specification of the possible behaviour of parallel and reactive systems.
To deal with reactive systems and guarantee their correct behavior in all possible environment, we need:

1. To study **mathematical models** for the formal description and analysis of concurrent programs.

2. To devise **formal languages** for the specification of the possible behaviour of parallel and reactive systems.

3. To develop **verification tools** and implementation techniques underlying them.
The basic Approach

The chosen abstraction for reactive systems is the notion of processes.

Systems evolution is based on process transformation: A process performs an action and becomes another process.

Everything is (or can be viewed as) a process. Buffers, shared memory, Linda tuple spaces, senders, receivers, ... are all processes.

Labelled Transition Systems (LTS) describe process behaviour, and permit modelling directly systems interaction.
Operational Semantics

To each process built using the above operators we associate an LTS by relying on structural induction to define the meaning of each operator.

Definition (Inference Systems)

An inference system is a set of inference rules of the form

\[
\begin{align*}
\ & p_1, \ldots, p_n \\
\rightarrow \\
\ & q
\end{align*}
\]

In our case for a generic operator \( op \) we shall have one or more rules like:

\[
\begin{align*}
E_{i_1} & \xrightarrow{\alpha_1} E'_{i_1} & \ldots & \ E_{i_m} & \xrightarrow{\alpha_m} & E'_{i_m} \\
\hline
\text{op}(E_1, \ldots, E_n) & \xrightarrow{\alpha} & \text{op}(E'_1, \ldots, E'_n)
\end{align*}
\]

where \( \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\} \).
Presentations of Labelled Transition Systems

Process Algebra as denotations of LTS

- LTS are represented by terms of process algebras.
- Terms are interpreted via operational semantics as LTS.

Process Algebra Basic Principles

1. Define a few elementary (atomic) processes modelling the simplest process behaviour;

2. Define appropriate composition operations to build more complex process behaviour from (existing) simpler ones.
A number of stochastic process algebras have been proposed in the last two decades. These are based on:

1. Labeled Transition Systems (LTS)
   - for providing compositional semantics of languages
   - for describing \textit{qualitative properties}

2. Continuous Time Markov Chains (CTMC)
   - for analysing \textit{quantitative properties}
A number of stochastic process algebras have been proposed in the last two decades. These are based on:

1. Labeled Transition Systems (LTS)
   - for providing compositional semantics of languages
   - for describing qualitative properties

2. Continuous Time Markov Chains (CTMC)
   - for analysing quantitative properties

Semantics of stochastic calculi have been provided by resorting to variants of the Structured Operational Semantics (SOS) approach but:

- there is no general framework for modelling the different formalisms
- it is rather difficult to appreciate differences and similarities of such semantics.
Stochastic Process Algebras - incomplete list

- TIPP (N. Glotz, U. Herzog, M. Rettelbach - 1993)
- Stochastic $\pi$-calculus (C. Priami - 1995, later with P. Quaglia)
- PEPA (J. Hillston - 1996)
- EMPA (M. Bernardo, R. Gorrieri - 1998)
- IMC (H. Hermanns - 2002)

... 

More Calculi will come: Besides qualitative aspects of distributed systems it more and more important that performance and dependability be addressed to deal, e.g., with issues related to quality of service.
Randomized Actions

- It is assumed that action execution takes time.
- Execution times is described by means of random variables.
- Random Variables are assumed to be exponentially distributed.
- Random Variables are fully characterised by their rates.
Common ingredients of Stochastic PA

**Randomized Actions**
- It is assumed that action execution takes **time**
- Execution times is described by means of **random variables**
- Random Variables are assumed to be **exponentially distributed**
- Random Variables are fully characterised by their **rates**.

**Properties of Exponential Distributions**

If $X$ is *exponentially distributed* with **parameter** $\lambda \in \mathbb{R}_{>0}$:

- $\mathbb{P}\{X \leq d\} = 1 - e^{-\lambda \cdot d}$, for $d \geq 0$
- The average duration of $X$ is $\frac{1}{\lambda}$; the variance of $X$ is $\frac{1}{\lambda^2}$
- **Memory-less**: $\mathbb{P}\{X \leq t + d \mid X > t\} = \mathbb{P}\{X \leq d\}$
Continuous Time Markov Chains

Continuous Time Markov Chains are a successful mathematical framework for modeling and analysing performance and dependability of systems that rely on exponential distribution of states transitions.

CTMCs come with
- Well established **Analysis Techniques**
  - **Steady State** Analysis
  - **Transient** Analysis
- Efficient **Software Tools** based on:
  - **Stochastic Timed/Temporal Logics**
  - **Stochastic Model Checking**
Continuous Time Markov Chains

Continuous Time Markov Chains are a successful mathematical framework for modeling and analysing performance and dependability of systems that rely on exponential distribution of states transitions.

CTMCs come with
- Well established Analysis Techniques
  - Steady State Analysis
  - Transient Analysis
- Efficient Software Tools based on:
  - Stochastic Timed/Temporal Logics
  - Stochastic Model Checking

A CTMC is a pair \((S, R)\)
- \(S\): a countable set of states
- \(R : S \times S \rightarrow R_{\geq 0}\), the rate matrix
A CTMC is associated to each process term;
CTMC model the stochastic behaviour of processes.
A CTMC is associated to each process term;
CTMC model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...
- compute *synchronizations rate*
A CTMC is associated to each process term; CTMC model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute *synchronizations rate* ...
- ...while taking into account transition multiplicity, for determining correct execution rate
A CTMC is associated to each process term; CTMC model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...
- compute *synchronizations rate* ...
- ...while taking into account transition multiplicity, for determining correct execution rate

**Process Calculi:**

\[
\alpha.P + \alpha.P = \alpha.P \\
\text{rec } X . \alpha.X | \text{rec } X . \alpha.X = \text{rec } X . \alpha.X
\]
Stochastic process calculi

- A CTMC is associated to each process term;
- CTMC model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...
- compute *synchronizations rate* ...
- ... while taking into account transition multiplicity, for determining correct execution rate

Stochastic Process Calculi:

\[
\alpha^\lambda.P + \alpha^\lambda.P \neq \alpha^\lambda.P \\
\text{rec } X \ . \ \alpha^\lambda.X \ | \ \text{rec } X \ . \ \alpha^\lambda.X \neq \text{rec } X \ . \ \alpha^\lambda.X
\]
A CTMC is associated to each process term; CTMC model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute synchronizations rate...
- ...while taking into account transition multiplicity, for determining correct execution rate

Stochastic Process Calculi:

\[ \alpha^\lambda P + \alpha^\lambda P = \alpha^{2\lambda} P \]

\[ \text{rec } X . \alpha^\lambda X \mid \text{rec } X . \alpha^\lambda X = \text{rec } X . \alpha^{2\lambda} X \]
We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone (FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.
We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone (FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.

Like most of the previous attempts we take a two steps approach: For a given term, say $T$, we define an enriched LTS and then use it to determine the CTMC to be associated to $T$. 
We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone (FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.

Like most of the previous attempts we take a two steps approach: For a given term, say $T$, we define an enriched LTS and then use it to determine the CTMC to be associated to $T$.

- Our variant of RTS associates terms and actions to functions from terms to rates.
- The *apparent rate* approach, originally developed by Hillston for multi-party synchronisation (à la CSP), is generalized to deal "appropriately" also with binary synchronisation (à la CCS).
Stochastic semantics of process calculi is defined by means of a transition relation $\rightarrow$ that associates to a pair $(P, \alpha)$ - consisting of process and an action - a total function $(\mathcal{P}, \mathcal{Q}, \ldots)$ that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.
Stochastic semantics of process calculi is defined by means of a transition relation that associates to a pair \((P, \alpha)\) - consisting of process and an action - a total function \((P, Q, \ldots)\) that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.

\[ P \xrightarrow{\alpha} P' \text{ means that, for a generic process } Q:\]

- if \(P(Q) = x \neq 0\) then \(Q\) is reachable from \(P\) via the execution of \(\alpha\) with rate/(weight) \(x\)
- if \(P(Q) = 0\) then \(Q\) is not reachable from \(P\) via \(\alpha\)
Stochastic semantics of process calculi is defined by means of a transition relation that associates to a pair \((P, \alpha)\) - consisting of process and an action - a total function \((P, Q, \ldots)\) that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.

\[ P \xrightarrow{\alpha} P \] means that, for a generic process \(Q\):
- If \(P(Q) = x \neq 0\) then \(Q\) is reachable from \(P\) via the execution of \(\alpha\) with rate/(weight) \(x\).
- If \(P(Q) = 0\) then \(Q\) is not reachable from \(P\) via \(\alpha\).

We have that if \(P \xrightarrow{\alpha} P\) then
- \(\oplus P = \sum_Q P(Q)\) represents the total rate/weight of \(\alpha\) in \(P\).
A rate transition system is a triple \((S, A, \rightarrow)\) where:

- \(S\) is a set of states;
- \(A\) is a set of transition labels;
- \(\rightarrow \subseteq S \times A \times [S \rightarrow \mathbb{R}_{\geq 0}]\)
Rate transition systems

Definition

A rate transition system is a triple \((S, A, \rightarrow)\) where:

- \(S\) is a set of states;
- \(A\) is a set of transition labels;
- \(\rightarrow \subseteq S \times A \times [S \rightarrow \mathbb{R}_{\geq 0}]\)

An example of RTS
Let $\mathcal{R} = (S, A, \rightarrow)$ be an RTS, then:

- $\mathcal{R}$ is **functional** if and only if for each $s \in S$, $\alpha \in A$, $P$ and $Q$ we have: $s \xrightarrow{\alpha} P$, $s \xrightarrow{\alpha} Q \implies P = Q$

- $\mathcal{R}$ is **image finite** if and only if for each $s \in S$, $\alpha \in A$ and $P$ such that $s \xrightarrow{\alpha} P$ we have: $\{s' | P(s') > 0\}$ is finite

A functional RTS

A general RTS

that leads to a CTMC.

that leads to a CTM Dec. Proc.
A translation from an RTS to a CTMC

An RTS:

\[ s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\beta} s_3 \xrightarrow{\gamma} s_4 \]

\[ \lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \]

The corresponding CTMC:

\[ s_1 \xrightarrow{\lambda_1 + \lambda_7} s_2 \xrightarrow{\lambda_4} s_3 \xrightarrow{\lambda_6} s_4 \]

\[ \lambda_2 \lambda_3 \lambda_5 \lambda_8 \lambda_7 \lambda_6 \lambda_5 \lambda_4 \lambda_3 \lambda_2 \lambda_1 \]
A translation from an RTS to a CTMC

An RTS:

The corresponding CTMC:
A translation from an RTS to a CTMC

An RTS:

The corresponding CTMC:
A translation from an RTS to a CTMC

An RTS:

The corresponding CTMC:
Strong Markovian Bisimilarity

**Definition (Bisimulation)**

Given a generic CTMC \((S, R)\)

- An equivalence relation \(\mathcal{E}\) on \(S\) is a Markovian bisimulation on \(S\) if and only if for all \((s_1, s_2) \in \mathcal{E}\) and for all equivalence classes \(C \in S/\mathcal{E}\) the following condition holds: \(R[s_1, C] = R[s_2, C]\).

**Definition (Bisimilarity)**

Given a generic CTMC \((S, R)\)

- Two states \(s_1, s_2 \in S\) are strongly Markovian bisimilar, written \(s_1 \sim_M s_2\), if and only if there exists a Markovian bisimulation \(\mathcal{E}\) on \(S\) with \((s_1, s_2) \in \mathcal{E}\).
Rate aware bisimulation

**Definition (Rate Aware Bisimilarity)**

Let $\mathcal{R} = (S, A, \rightarrow)$ be a RTS:

- An equivalence relation $\mathcal{E} \subseteq S \times S$ is a *rate aware* bisimulation if and only if, for all $(s_1, s_2) \in \mathcal{E}$, and $S' \in S/\mathcal{E}$, and for all $\alpha$ and $\mathcal{P}$:
  
  $s_1 \xrightarrow{\alpha} \mathcal{P} \Rightarrow \exists \mathcal{Q} : s_2 \xrightarrow{\alpha} \mathcal{Q} \land \mathcal{P}(S) = \mathcal{Q}(S)$

- Two states $s_1, s_2 \in S$ are *rate aware bisimilar* ($s_1 \sim s_2$) if there exists a rate aware bisimulation $\mathcal{E}$ such that $(s_1, s_2) \in \mathcal{E}$.

**Theorem**

Let $\mathcal{R} = (S, A, \rightarrow)$, for each $A' \subseteq A$ and for each $s_1, s_2 \in S$ and $(S, \mathcal{R}) = CTMC[\{s_1, s_2\}, A']$: $s_1 \sim s_2 \implies s_1 \sim_M s_2$

*Rate aware bisimilarity* and *strong bisimilarity* coincide if actions identity is not considered.
PEPA: Performance Process Algebra

PEPA Systems

PEPA systems are the result of *components* interaction via *activities*:
- Components reflect the behaviour of relevant parts of the system,
- activities model the actions components do perform.

PEPA Activities

Each PEPA activity consists of a pair \((\alpha, \lambda)\) where:
- \(\alpha\) symbolically denotes the performed action;
- \(\lambda > 0\) is the rate of the (negative) *exponential* distribution.

PEPA Syntax

If \(\mathcal{A}\) is a set of *actions*, ranged over by \(\alpha, \alpha', \alpha_1, \ldots\), then \(\mathcal{P}_{PEPA}\) is the set of process terms \(P, P', P_1, \ldots\) defined by:

\[
P ::= (\alpha, \lambda).P \mid P + P \mid P \parallel L P \mid P/L \mid A
\]
PEPA Stochastic semantics...
Prefixes and Sums

\[
\begin{align*}
(\alpha, \lambda).P & \xrightarrow{\alpha} [P \leftrightarrow \lambda] \quad \text{(Act)} \\
(\alpha, \lambda).P & \xrightarrow{\beta} \emptyset \quad \text{(\emptyset-Act)} \\
P & \xrightarrow{\alpha} P \quad Q & \xrightarrow{\alpha} Q \\
\sum P + Q & \xrightarrow{\alpha} P + Q
\end{align*}
\]
PEPA Stochastic semantics...

Prefixes and Sums

\[(\alpha, \lambda).P \xrightarrow{\alpha} [P \mapsto \lambda] (\text{Act})\]

\[\alpha \neq \beta \quad (\emptyset\text{-Act})\]

\[P \xrightarrow{\alpha} P \quad Q \xrightarrow{\alpha} Q\]

\[P + Q \xrightarrow{\alpha} P + Q \quad (\text{SUM})\]

Interleaving and Multiparty Synchronization

\[\begin{align*}
P & \xrightarrow{\alpha} P \\
Q & \xrightarrow{\alpha} Q \\
\alpha & \notin L
\end{align*}\]

\[\begin{align*}
P & \parallel_L Q \xrightarrow{\alpha} P \parallel_L \chi_Q + \chi_P \parallel_L Q
\end{align*}\]

\[\begin{align*}
P & \xrightarrow{\alpha} P \\
Q & \xrightarrow{\alpha} Q \\
\alpha & \in L
\end{align*}\]

\[\begin{align*}
P & \parallel_L Q \xrightarrow{\alpha} P \parallel_L Q \cdot \frac{\min\{\oplus P, \oplus Q\}}{\oplus P \oplus Q}
\end{align*}\]
Other Stochastic Process Algebras considered

- EMPA
- TIPP
- Stochastic CCS
- Stochastic $\pi$-calculus
- IMC
- STOKLAIM
- MarCaSPiS

Tools

For all these formalisms we have built prototypes to perform systems analysis
In the last twenty years we have seen many variants of process algebras that have been introduced to capture different non-qualitative aspects of concurrent systems:

- probability
- Time
- Causality
- Security
- ...
In the last twenty years we have seen many variants of process algebras that have been introduced to capture different non-qualitative aspects of concurrent systems:

- probability
- Time
- Causality
- Security
- ...
Let $D$ be a complete partial order (cpo) with least element $\bot$, a Uniform Transition System on $D$, that we call $D – ULTRAS$, is a triple $(S, A, \rightarrow)$ where:

- $S$ is a set of states,
- $A$ a set of transition labels,
- $\rightarrow$ a subset of $S \times A \times [S \rightarrow D]$. 

ULTRAS: Uniform Labelled Transition Systems
Traces:

Let $\mathcal{U} = (S, A, \rightarrow)$ be a $D$-ULTRAS. A trace $\alpha$ for $\mathcal{U}$ is a finite sequence of transition labels in $A^*$, where $\alpha = \varepsilon$ denotes the empty sequence while operation “$\cdot \circ \cdot$” denotes sequence concatenation.

Measuring function:

Let $\mathcal{U} = (S, A, \rightarrow)$ be a $D$-ULTRAS and $M$ be a lattice. A measuring function for $\mathcal{U}$ is a function $\mathcal{M}_M : S \times A^* \times 2^S \rightarrow M$. 
Let $\mathcal{U} = (S, A, \rightarrow)$ be a $D$-ULTRAS and $\mathcal{M}_M$ be a measuring function for $\mathcal{U}$. We say that $s_1, s_2 \in S$ are $\mathcal{M}_M$-trace equivalent, written $s_1 \sim_{\text{Tr}, \mathcal{M}_M} s_2$, iff for all traces $\alpha \in A^*$:

$$\mathcal{M}_M(s_1, \alpha, S) = \mathcal{M}_M(s_2, \alpha, S)$$
Let $\mathcal{U} = (S, A, \rightarrow)$ be a $D$-ULTRAS and $\mathcal{M}_M$ be a measuring function for $\mathcal{U}$. An equivalence relation $\mathcal{B}$ over $S$ is an $\mathcal{M}_M$-bisimulation iff, whenever $(s_1, s_2) \in \mathcal{B}$, then for all traces $\alpha \in A^*$ and equivalence classes $C \in S/\mathcal{B}$:

$$\mathcal{M}_M(s_1, \alpha, C) = \mathcal{M}_M(s_2, \alpha, C)$$

We say that $s_1, s_2 \in S$ are $\mathcal{M}_M$-bisimilar, written $s_1 \sim_{B, \mathcal{M}_M} s_2$, iff there exists an $\mathcal{M}_M$-bisimulation $\mathcal{B}$ over $S$ such that $(s_1, s_2) \in \mathcal{B}$. 
Behavioural equivalences for ULTRAS

Testing equivalence:

Let $(S, A, \rightarrow)$ be a $D$-ULTRAS and $M_M$ be a measuring function for $\mathcal{U}$.

Two states $s_1, s_2$ are testing equivalent if and only if, for each trace $\alpha \in A^*$ and $A' \subseteq A$:

$$M_M(s_1, \alpha, M_{set}(A')) = M_M(s_2, \alpha, M_{set}(A'))$$

Must Sets:

Let $R = (S, A, \rightarrow)$ be a $D$-ULTRAS, $A' \subseteq A$, $s \in S$ and $a \in A$:

- $s$ must $a$ iff $\exists P \neq \lambda x.\bot$ such that $s \xrightarrow{a} P$
- $s$ Must $A'$ iff $\exists a \in A'$ such that $s$ must $a$
- $M_{set}(A') = \{s \in S | s$ Must $A'\}$
Labelled Transition System:

A labeled transition system (LTS for short) is a triple \((S, A, \rightarrow)\) where:

- \(S\) is a countable set of states.
- \(A\) is a countable set of transition-labeling actions.
- \(\rightarrow \subseteq S \times A \times S\) is a transition relation.

A Labelled Transition System can be rendered as a \(\mathbb{B}\)-ULTRAS where:

- \(\mathbb{B}\) is the set of boolean values \(\{\top, \bot\}\);

Measuring function:

\[
M_{\mathbb{B}}(s, \alpha, S') = \begin{cases} 
1 & \text{if } s \text{ reaches } S' \text{ with } \alpha \\
0 & \text{otherwise}
\end{cases}
\]
An action-labeled discrete-time Markov chain (ADTMC for short) is a triple \((S, A, \rightarrow)\) where:

- \(S\) is a countable set of states.
- \(A\) is a countable set of transition-labeling actions.
- \(\rightarrow \subseteq S \times A \times \mathbb{R}_{(0,1]} \times S\) is a (probabilistic) transition relation.

ADTMC is a functional \(\mathbb{R}_{[0,1]}\)-ULTRAS in which:

\[
\sum_s \sum_{a \in A} \sum_{s' \in S} D(s') \in \{0, 1\}
\]

Measuring function:

\[
\mathcal{M}_{[0,1]}(s, \alpha, S') = \text{probability to reach a state in } S' \text{ from } s \text{ with trace } \alpha.
\]
An action-labeled continuous-time Markov chain (ACTMC for short) is a triple \((S, A, \rightarrow)\) where:

- \(S\) is a countable set of states.
- \(A\) is a countable set of transition-labeling actions.
- \(\rightarrow \subseteq S \times A \times \mathbb{R}_{>0} \times S\) is a transition relation.

An ACTMC is a functional \(R_{\geq 0}\)-ULTRAS.

**Measuring function:**

\[
\mathcal{M}_{R_{\geq 0}[0,1]}(s, \alpha, S') = \text{probability distribution to reach a state in } S' \text{ from } s \text{ with trace } \alpha.
\]
Trace, Testing and Bisimulation equivalences classically defined on LTS, ADTMC and ACTMC coincide with the ones induced by:

- $\mathbb{B}$ – ULTRAS and $\mathcal{M}_\mathbb{B}$;
- $[0, 1]$ – ULTRAS and $\mathcal{M}_{[0,1]}$;
- $\mathbb{R}_{\geq 0}$ – ULTRAS and $\mathcal{M}_{\mathbb{R}_{\geq 0} \to [0,1]}$. 

R. De Nicola (DSIUF)
We have:

- introduced Rate Transition Systems and have used them as the basic model for defining stochastic behaviours of processes.
- introduced a natural notion of bisimulation over RTS that agrees with Markovian bisimulation.
- shown how RTS can be used to provide the stochastic operational semantics of PEPA (... and other SPA).
- introduced ULTRA\(S\)s as more general models of quantitative systems
- defined equivalence relations over ULTRA\(S\)
- shown that ULTRA\(S\) can be used for modelling other semantics (non-deterministic, stochastic, probabilistic, ... )
Thank you for your attention!