Deriving Performance Models from Software Architecture Specifications

Получение модели производительности из спецификаций архитектуры ПО

Simonetta Balsamo, Marta Simeoni

Аннотация

В настоящее время оценена важность и полезность количественного анализа ПО для оценки требований ресурсов, и разработки систем в целом. Особенно важно, что анализ производительности, наконец, включается в жизненный цикл ПО на ранних стадиях. С этой целью предлагается несколько различных подходов, объединяющих анализ производительности и спецификации ПО (СПО). В статье представлен краткий обзор и сравнение некоторых подходов для получения моделей производительности по СПО. Уделяется внимание применимости предлагаемых методов, на требуемые ограничения и допущения, тип модели производительности, разделение модели производительности и языка спецификации, их реализацию и то, насколько легко разработчику ПО понять полученные результаты производительности.

1. Введение

Растет интерес к количественному анализу ПО, поскольку это чрезвычайно важно для оценки требований ресурсов, и разработки систем в целом. Особенно важно, что анализ производительности, наконец, включается в жизненный цикл ПО на ранних стадиях. [Smi90, WOSP2000]. Под спецификацией ПО (СПО) в данной статье понимается структура ПО, как верхний уровень абстракции, предполагаемый для использования на соответствующем уровне разработки для выполнения количественного анализа. С этой целью недавно предложены несколько подходов, объединяющих анализ производительности и СПО. Рассмотрены несколько типов моделей производительности и различных языков спецификации.

В статье рассматривается несколько подходов, которые позволяют получить модель производительности по СПО. Представлено краткое описание методов и сравнение каждого подхода. Уделяется внимание применимости предлагаемых методов, требуемые ограничения и допущения, тип модели производительности, разделение модели производительности и языка спецификации, их реализацию. То, насколько легко разработчику ПО понять полученные результаты производительности при проведении анализа, является еще одним важным вопросом в рассмотрении различных методов. Несколько методов основано на СПО, базирующейся на языке UML, который становится стандартом описания для СПО [BRJ99]. Он имеет несколько типов диаграмм, которые позволяют описать различные свойства системы, например статические характеристики и поведение системы, взаимодействие системных компонент и особенности физической реализации.

Статья организована следующим образом: раздел 2 содержит описание различных подходов для получения модели производительности по СПО, в разделе 3 представлено сравнение методов, в разделе 4 представлено заключение.

2. От СПО к модели производительности. Некоторые подходы.

В данном разделе представлено краткое описание некоторых методов для преобразования СПО в модель производительности. Принятая система обозначения этих методов использует для образования названия метода инициалы авторов метода.

Рассматривается применимость методов, а также требуемые ограничения и допущения к моделе ПО. Каждый из методов базируется на определенном типе модели производительности и языке описания. Язык описания включает формальные описания, типа алгебр процессов, сетей Петри и химической абстрактной машины и языки, основанные на UML. Модель производительности включает сети массового обслуживания (CMO, QN), их расширение, называемое расширенные CMO (EQN), многоуровневые CMO (LQN), стохастические временные сети Петри (STPN), стохастические алгебры процессов (SPA) и имитационные модели.

Некоторые из предлагаемых методов основаны на методе, называемом технология расчета производительности ПО (Software Performance Engineering - SPE), предложенном Смитом [Smi90]. Он впервые предложил метод, позволяющий осуществить полноценную интеграцию анализа производительности в процесс разработки ПО от ранних стадий проекта до его завершения. SPE базируется на двух моделях – модель выполнения ПО и модель работы системы. Первая базируется на графе выполнения и представляет характер поведения ПО. Вторая базируется на модели CMO и представляет модель вычислительной системы, включающую аппаратные и программные компоненты. Анализ модели ПО дает информацию о необходимых ресурсах вычислительной системы. Полученные результаты, вместе с информацией об устройствах являются входными для модели работы системы, которая представляет модель всей аппаратно/программной системы.

2.1 Основные методы

Рассмотрим основные методы преобразования СПО в модель производительности.

WS: Вильямс и Смит в работе [WS98] применяют метод SPE для оценки характеристик производительности СПО. Основной акцент в работе делается на построение и анализ модели выполнения ПО, которая является целевой из СПО и получается из диаграмм последовательности. Диаграммы классов и диаграммы размещения используются для дополнения СПО, но не включаются в процесс преобразования. Процесс SPE требует дополнительной информации, которая включает требования к программным ресурсам для выполнения, а также конфигурационные данные вычислительной системы.

MG: В работе [MG98], Менаскэ и Гома представили метод для получения модели QN, разработанный и использованный авторами в разработке клиент/серверных приложений. Метод базируется на языке вычисления производительности клиент-серверных архитектур (CLIent/Server Software Performance Evaluation - CLISSPE) [M97]. Метод незначительно использует возможности UML. Функциональные требования к системе определяются в терминах вариантов использования, а модель системы определяется аналогично диаграммам классов. Диаграмма вариантов использования вместе с клиент/серверной СПО и схемой отображения программных компонентов на аппаратные устройства используется для разработки CLISSPE-спецификации программы. Система CLISSPE обеспечивает формирование соответствующей модели QN. Возможность получение модели LQN рассматривается в работах [RS95, WNPN 95].

BIM: В работе [Bim98] Бальзамо и др. предлагают метод автоматизированного получения модели QN из СПО, описанной с использованием терминов химической абстрактной машины (CHemical Abstract Machine - CHAM) [IW95]. Коротко, в CHAM, СПО

представляется посредством множества молекул, представляющих собой статические компоненты системы, множество правил реакций представляет собой динамическое выполнение системы посредством единичных реакций, а начальный раствор представляет начальные условия. В статье представлен алгоритм получения QN-модели из CHAM СПО. Он основан на механизме размеченных переходов системы, который представляет поведение системы в CHAM. Алгоритм не полностью определяет модель QN. Такие параметры, как распределение времени обслуживания и параметры входных потоков потребителей определяются разработчиком. Решение модели QN возможно как аналитическое, так и в символьном виде. Параметр реализации определяет возможный сценарий выполнения, а полученные характеристики производительности позволяют оценить, насколько полученные характеристики разрабатываемого процесса удовлетворяют заданным.

CM: В работе [CM00] Кортелес и Мирандола предлагают метод, основанный на объединении информации из различных диаграмм UML – диаграмм размещения, последовательности и вариантов использования. Данный метод является формальным расширением подхода, предложенным в [WS98] и состоит из следующих шагов:

- В диаграмме вариантов использования добавляется вероятность для каждой связи вариант использования – пользователь. В дальнейшем они используются в соответствующих диаграммах последовательностей.
- Для каждого варианта использования, по соответствующим ему диаграммам последовательности, строится частный граф выполнения (meta-EG). Далее, последовательно, по мере обработки вариантов использования, строится объединенный граф выполнения. Рассматриваются только те диаграммы последовательности, которые не содержат асинхронных взаимодействий.
- Используя диаграммы размещения, получают модель EQN для аппаратной платформы, а также, после сшивания отдельных meta-EG, получают экземпляр графа размещения, определяющего рабочую нагрузку EQN. Основная идея на этом шаге заключается в том, чтобы использовать диаграммы размещения, которые отображают топологию платформы и типы объектов, имеющих необходимую для построения EQN информацию, например внутренняя структура и параметры устройств. Более того, частные EG определяют тип взаимодействия и размер данных для обмена. В статье не представлены детали построения EQN графа по диаграмме размещения.
- В частный EQ граф конструктор добавляет необходимые числовые параметры.
- Частные EQ графы объединяются с графом EQNM для решения модели производительности по методу SPE. Следует заметить, что ключевой момент в этой работе – это добавление необходимой для вычисления производительности информации в соответствующие диаграммы UML и получение EQN-модели. ППО для данного подхода еще не реализовано.

AABI: Группа Андолфи [AABI00] предлагает метод автоматического получения QN-модели по СПО, представленной в виде диаграмм обмена сообщениями (MSC), которые соответствуют диаграммам последовательности в терминологии UML. Принцип заключается в том, чтобы анализировать MSC в терминах языков трассировки (последовательность событий), на которых представлены формируемые последовательности событий при необходимой степени параллелизма и динамических взаимозависимостей. Эта информация используется для построения соответствующей QN-модели. В статье представлен алгоритм её получения. Данный подход базируется на предыдущем, изложенном в работе [BIM98], который был предложен с целью снижения высокой вычислительной сложности решения CHAM-модели, возникающей вследствие лавинного роста числа конечных состояний вероятностной модели.

ABI: Подход, предложенный группой Аквелани [ABI00] заключается в получении QN-модели из системы размеченных переходов (LTS), описывающей поведение СПО. Используя LTS становится возможным абстрагировать модель от любых особенностей языка СПО. Подход предполагает, что LTS является единственным необходимым источником данных о системе, который можно использовать. Подход предполагает конечное число состояний, независящих от языка СПО и позволяет моделировать более сложные механизмы взаимодействия и ограничения при синхронизации процессов, что может быть отображено моделью EQN. Такие EQN-модели отображают параллельное выполнение и взаимодействие компонентов на уровне разработки СПО.

BCD: Группа Бернандо [BCD00], как средство описания архитектуры системы, предлагает язык, основанный на стохастических временных алгебрах процессов. Этот подход обеспечивает единство формального языка описания и модели производительности. Его целью является описание и анализ как функциональных, так и временных характеристик в структуре СПО. Подход предлагает адоптацию языка описания структуры системы, называемого ÆMPA, дав ему синтаксис и семантику спецификации EMPA, то есть стохастической алгебры процессов. Авторы иллюстрируют различные функциональные и нефункциональные характеристики, включая вычисление производительности, основываясь на формировании марковских цепей, которые решаются численно. Авторы предлагают использование ППО TwoTower [B00] в качестве инструмента для системного моделирования и анализа функциональных и временных характеристик, поскольку данный пакет поддерживает описание EMPA.

2.2 Методы, основанные на конструктивных шаблонах

Конструктивные шаблоны представляют собой часто используемые решения и используются для описания СПО. Каждый шаблон определяется своей собственной структурой (сущность компонента) и поведением (как компонент взаимодействует с другими компонентами). Некоторые подходы рассматривают СПО как совокупность конструктивных шаблонов и получают из них соответствующую модель производительности. Они используют спецификацию UML. Такие подходы однозначно определяют соответствие между каждым шаблоном и его моделью производительности.

GM: В работе [GM00] Гома и Менасце исследуют разработку и моделирование производительности шаблонов взаимодействия компонентов клиент-серверной системы. Такие шаблоны определяют способ взаимодействия клиентских и серверных компонентов посредством специальных соединителей. Идея заключается в том, чтобы начать разработку шаблонов взаимодействия компонентов посредством UML, используя диаграммы классов (для моделирования статических характеристик) и диаграмм взаимодействия (для описания взаимодействия между компонентами и соединяющих объектов – экземпляров классов, описываемых диаграммами классов). Это модели, снабженные дополнительной информацией о времени выполнения, для того, чтобы обеспечить объединение структурных и динамических параметров, преобразуются в XML-описание. Модели производительности рассматриваемых шаблонов представляют собой EQN-модели. Их вид зависит от типа взаимодействия, что описано в предыдущей работе авторов. EQN-модели преобразуют в марковские цепи либо решают приближенными аналитическими методами.

PW: В работах [PW99, P00] Петриу и Ванг рассматривают значительный набор конструктивных шаблонов (каналы и фильтры, клиенты и серверы, трансляторы запросов, уровни, критические секции) описанных с использованием UML-диаграмм

взаимодействия, которые объединяются с диаграммами классов и диаграммами последовательности, выделяя взаимодействующие объекты. В работе показано соответствие рассматриваемых моделей LQN-моделям. Более того, авторы предлагают способ построения СПО, посредством объединения соответствующих шаблонов. Предлагаемый подход руководствуется методом SPE и формирует модели выполнения системы и программы, основываясь на методе преобразования графа. СПО определяется при помощи UML- диаграмм взаимодействия, размещения и вариантов использования. Диаграммы последовательностей, как часть диаграмм взаимодействия, используются для построения модели выполнения программы (представляющей собой UML-диаграмму деятельности); диаграмма классов используется для построения модели работы системы (представляющей собой LQN-модель). Диаграмма вариантов использования предоставляет информацию о рабочей нагрузке, а диаграмма размещения – о размещении программных и аппаратных компонентов.

2.3 Имитационный метод

Рассмотрим два подхода, основанных на имитационных моделях, которые рассматривают имитационные блоки для определения модели, структура и входные параметры которой получены из UML-диаграмм.

AS: Подход, предлагаемый Арифом и Шпайрсом в работе [AS00] представляет собой имитационную структуру, названную имитационным языком моделирования (SimML), которая формирует имитационную программу по соответствующей UML-спецификации. Предлагаемый пакет построения UML-диаграмм позволяет пользователю нарисовать диаграммы классов и последовательностей, описать необходимую информацию и автоматически сформировать ориентированную на программный процесс имитационную модель. Имитационная программа формируется на языке программирования Java. Подход предлагает XML-преобразование исходной UML-модели для того, чтобы обеспечить одновременное хранение структурной информации и информации, необходимой для имитационного моделирования.

DLHBP: Подход, предложенный группой Мигеля [DLHBP00], фокусируется на системах реального времени и предлагает расширение UML-диаграмм для отображения временных характеристик и использования ресурсов. Расширение основано на использовании стереотипов – связных значениях и ограничениях стереотипами. СПО представляет собой расширенные UML-диаграммы без ограничения на использования их типа. Затем UML-модель используется для автоматического формирования соответствующей имитационной модели OPNET. Диаграммы UML также определяют промежуточную модель для программного анализа. Имитационная модель определяется реализацией программы, представленной различными UML-метаклассами. Подход формирует механизм обратной связи – после анализа и имитации модели, некоторые результаты включаются в связные значения элементов UML-диаграмм. Это обеспечивает конструктора СПО наглядной обратной связью исходной модели и результатами производительности.

2.4. Преобразование моделей в конкретных ситуациях

Некоторые подходы рассматривают получение модели производительности из СПО на конкретных примерах. Они рассматривают UML-спецификацию и различные типы моделей производительности.

KP: В работе [KP99] Кинг и Пули показали на примерах, как сформировать модель стохастической временной сети Петри из UML-спецификации. Они рассматривают диаграмму вариантов использования и объединенные диаграммы взаимодействия и

состояний всех взаимодействующих объектов и соответствующих вложенных объектов. Основная идея заключается в том, чтобы преобразовать каждую диаграмму состояний, которая представляет объекты диаграммы взаимодействия в сеть Петри. Полученное множество сетей Петри может быть объединено в модель стохастической временной сети Петри, представляющей всю систему в целом, а именно обобщенную стохастическую сеть Петри (GSPN) [ABC86]. Объединение сетей рассмотрено только на примерах, то есть в статье не приводится процедура их объединения. Сеть GSPN может быть проанализирована при помощи ППО SPNP.

PK: В работе [PK99] Пули и Кинг описывают некоторые предварительные идеи относительно получения QN-моделей из UML-спецификации. Диаграммы вариантов использования используются для назначения параметров рабочей нагрузки и моделирования различных классов запросов системы. Диаграммы реализации используются для определения использования и доступности системных ресурсов. Основная идея заключается в том, чтобы определить соответствие объединенных диаграмм размещения и компонентов с моделями QN, отображающим компоненты и их связи на центры обслуживания.

P: В работе [Poo99] Пули описывает получение модели в терминах стохастической алгебры процессов из спецификации UML. Начало метода аналогично предложенному в работе [KP99], т.е. СПО описывается посредством объединения диаграмм взаимодействия и диаграмм состояния всех взаимодействующих и вложенных в них объектов. Основная идея метода – получение описания в терминах стохастической алгебры процессов для каждого объекта диаграммы взаимодействия и их объединение в одну модель. В статье показано, как это может быть выполнено на одном простом примере. В статье также представлена попытка формирования марковской цепи с непрерывным временем непосредственно из диаграмм UML. Ключевой момент здесь заключается в том, что в любой момент времени для каждого объекта на диаграмме взаимодействия, существует только одно внутреннее состояние. Совокупность объектов в текущем состоянии называется маркировкой. Получение всех возможных маркировок возможно посредством проведения взаимодействий – это позволяет построить соответствующую диаграмму переходов и, затем, марковские цепи.

SW: Смит и Вильямс в работе [SW97] представляют пример, иллюстрирующий получение модели производительности из объектно-ориентированной модели проектирования и предлагают использование ППО SPE·ED, поддерживающего метод SPE для вычисления характеристик объектно-ориентированных систем. Начиная с множества вариантов, описанных диаграммами последовательности сообщений, они получают граф выполнения, который определяет модель выполнения ПО. QN-модель анализируется приближенными аналитическими или имитационными методами, реализованными в SPE·ED. Работа соответствует рассмотренному в главе 2.1 подходу WS.

H: В работе [Hoe00] Хобен рассматривает некоторые правила, которые могут быть использованы для вычисления, либо добавления информации, полезной для выполнения расчета производительности в различных UML-диаграммах. В работе предлагаются некоторые UML-расширения, основанные на использовании стереотипов и связных значений, а также некоторых правил распространения запросов пользователей, определяемых UML-моделями для получения модели производительности. Эти правила позволяют выполнить оценку производительности UML-моделей на различных уровнях абстракции и формируют прототип ППО, способного автоматически сформировать оценки производительности, основываясь на использовании QN-моделей. Автор дает

только некоторые моменты получения QN-модели из UML, не предоставляя полного описания.

Таблица 1. Обобщение некоторых схожих особенностей рассматриваемых подходов.

Метод	***Язык СПО***	***Ограничения***	***Модель производительности***
WS	MSC-UML: диаграммы размещения, последовательности и классов		EQN
MG	Клиент-серверные спецификации, варианты использования	Клиент-серверные системы	QN/LQN
BIM	CHAM		QN
CM	UML-диаграммы размещения, последовательности и вариантов использования		EQN
AABI	MSC-UML: диаграммы последовательности		QN
ABI	LTS		EQN
BCD	ÆMPA		SPA
GM	MSC-UML: диаграммы взаимодействия и классов	Клиент-серверные системы; шаблоны взаимодействия для синхронных, асинхронных взаимодействий с многопоточным сервером	EQN
PW	UML-диаграммы размещения, вариантов использования и деятельности	Шаблоны для клиент-серверных систем, каналы и фильтры, передатчики запросов, критические секции	LQN
AS	UML-диаграммы последовательности и классов		Имитация
DLHBP	Все UML-диаграммы	Системы реального времени	Имитация
KP	UML-диаграммы вариантов использования, объединенные диаграммы состояний и взаимодействия		STPN(GSPN)
PK	UML-диаграммы вариантов использования, объединенные диаграммы размещения и компонентов		QN
P	Объединенные UML-диаграммы состояний и взаимодействия		SPA
SW	MSC		EQN
H	Все UML-диаграммы		QN

3. Обобщение

Различные подходы, описанные в предыдущей главе выстроены от основных направлений до отдельных примеров. Рассмотрены различные типы моделей производительности и различные языки определения.

Большинство подходов претендует на полный жизненный цикл ПО, тогда как некоторые только на этап проектирования. Последние обычно рассматривают формальное моделирование поведения программы, основываясь на стохастические сети Петри, либо на стохастические алгебры процессов, как, например, в BCD. Такие модели объединяют функциональные и нефункциональные характеристики, а также обеспечивают однозначное соответствие СПО и модели производительности. Однако, с точки зрения вычисления производительности, методы их анализа часто ссылаются на использование марковских цепей, которые легко могут вызвать проблему вычисления, вследствие лавинного роста числа состояний. Более того, с точки зрения проектирования ПО, это далеко не простой способ отображения результатов производительности на уровне СПО.

Несколько подходов основано на моделях сетей массового обслуживания (QN, EQN, LQN) и оперируют с верхним уровнем абстракции, что позволяет им иметь более простую обратную связь с СПО, чем более детализированные стохастические модели. Особенно, если СПО компоненты могут соответствовать компонентам сети массового обслуживания (например центры обслуживания или подсети), результаты производительности могут быть прямо отображены на уровень СПО.

Важная особенность подходов WS и AABI заключается в том, что они используются для анализа СПО без определения аппаратной платформы. Однако, поскольку подход WS (и CM) использует EQN-модель, полученную по рекомендации SPE, подход определяет QN-модель посредством анализа системы размеченных переходов, соответствующей диаграмме последовательности.

Некоторые подходы вводят ограничения на СПО, например MG используется для клиент-серверных систем или GM и PW, которые рассматривают конкретные конструктивные шаблоны. Однако два последних содержат способ объединения этих шаблонов.

Большинство рассмотренных подходов рассматривает UML-спецификацию и различные ее диаграммы как основу для получения и параметризации модели. Часто считается необходимым введение дополнительных параметров в UML-диаграммы посредством простых надписей или расширений, основанных на стереотипах и связных значений для выполнения полной спецификации программной модели и дальнейшего получения модели производительности.

Различные ППО предлагаются или используются для выполнения некоторых шагов рассмотренных подходов. Однако ни один из них не содержит полной системы описания, анализа производительности и обратной связи с разработчиком ПО. Открытой проблемой остается полная автоматизация процесса получения модели производительности и интеграция соответствующих ППО.

4. Заключение

Объединение СПО и моделей производительности является важной задачей на ранних этапах жизненного цикла ПО. Рассмотрены различные подходы получения моделей производительности из СПО, с указанием их общих особенностей, типа модели производительности и языка описания, реализации и обратной связи с разработчиком ПО.

Библиография

[ABC86] M. Ajmone, G. Balbo, G. Conte "Performance Models of Multiprocessor Performance" The MIT Press (1986).
[AABI00] F. Andolfi, F. Aquilani, S. Balsamo, P. Inverardi "Deriving Performance Models of Software Architectures from Message
Sequence Charts" in [WOSP2000].
[ABI00] F. Aquilani, S. Balsamo, P. Inverardi "Performance Analysis at the software architecture design level" Technial Report TR-SAL-32, Technical Report Saladin Project, 2000, submitted for pubblication.
[AS00] L.B. Arief, N.A. Speirs "A UML Tool for an Automatic Generation of Simulation Programs" in [WOSP2000] pp. 71-76.
[BIM98] S. Balsamo, P. Inverardi, C. Mangano "An Approach to Performance Evaluation of Software Architectures" Workshop on Software and Performance, WOSP'98, Santa Fe, New Mexico, Oct 12-16, 1998.
[B00] M. Bernardo "Theory and Application of Extended Markovian Process Algebra" Ph. D. Thesis, University of Bologna (Italy), 1999.
[BCD00] M. Bernardo, P. Ciancarini, L. Donatiello "AEMPA: A Process Algebraic Description Language for the Performance Analysis of Software Architectures" in [WOSP2000].
[BRJ99] G. Booch, J. Rumbaugh, I. Jacobson "The Unified Modeling Language User Guide" Addison-Wesley (1999).
[CM00] V. Cortellessa, R. Mirandola "Deriving a Queueing Network based Performance Model from UML Diagrams" in [WOSP2000] pp. 58-70.
[DLHBP00] M. De Miguel, T. Lambolais, M. Hannouz, S. Betge-Brezetz, S. Piekarec "UML Extensions for the Specification and Evaluation of Latency Constraints in Architectural Models" in [WOSP2000] pp. 83-88.
[GM00] H. Gomaa, D.A. Menasce "Design and Performance Modeling of Component Interconnection Patterns for Distributed Software Architectures" in [WOSP2000] pp. 117-126.
[Hoe00] F. Hoeben "Using UML Models for Performance Calculation" in [WOSP2000] pp. 77-82.
[IW95] P. Inverardi, A.L. Wolf "Formal Specification and Analysis of Software Architectures Using the Chemical Abstract Machine Model" IEEE Transaction on Software Engineering, Vol. 21, n. 4, pp. 373-386 (1995).
[Kan92] K. Kant "Introduction to Computer System Performance Evaluation" McGraw-Hill (1992).
[KP99] P. King, R. Pooley "Derivation of Petri Net Performance Models from UML Specifications of Communication Software" Proc. of XV UK Performance Engineering Workshop (1999).
[M97] D.A. Menasce "A Framework for Software Performance Engineering of Client/Server Systems" Proc. of the 1997 Computer Measurement Group Conference, Orlando, Florida (1997).
[MG98] D.A. Menasce, H. Gomaa "On a Language Based Method for Software Performance Engineering of Client/Server Systems" Proc. of WOSP'98, Santa Fe, New Mexico, USA, pp. 63-69 (1998).
[P00] D. Petriu "Deriving Performance Models from UML Models by Graph Transformations" Tutorial in [WOSP2000].
[PW99] D. Petriu, X. Wang "From UML descriptions of High-Level Software Architectures to LQN Performance Models" in Proc. of AGTIVE'99, Springer Verlag LNCS 1779, pp. 47-62 (1999).
[PK99] R. Pooley, P. King "The Unified Modeling Language and Performance Engineering" in Proc. of IEE Software (1999).
[Poo99] R. Pooley "Using UML to Derive Stochastic Process Algebra Models" Proc. of XV UK Performance Engineering Workshop (1999).
[RS95] J.A. Rolia and K.C. Sevcik "The Method of Layers" IEEE Trans. on Software Eng., Vol. 21/8, pp. 682-688 (1995).
[SG96] M. Shaw, D. Garlan "Software Architecture: Perspectives on an Emerging Discipline" Prentice Hall (1996).
[Smi90] C.U. Smith, "Performance Engineering of Software Systems" Addison Wesley (1990).
[SW97] C.U. Smith, L.G. Williams "Performance Engineering Evaluation of Object Oriented Systems with SPE•ED" in 'Computer Performance Evaluation: Modelling Techniques and Tools' LNCS 1245 (R. Marie et alt. Eds.), Springer-Verlag, 1997.
[WS98] L.G. Williams, C.U. Smith "Performance Evaluation of Software Architectures" in Proc. of WOSP'98, Santa Fe, New Mexico, USA, pp. 164-177 (1998).
[WNPM95] C. Woodside, J. Neilson, S. Petriu, S. Mjumdar "The Stochastic rendezvous network model for performance of synchronous client-server-like distributed software" IEEE Transaction on Computer, Vol. 44, pp. 20-34 (1995).
[WOSP2000] ACM Proceedings of Workshop on Software and Performance, WOSP2000, Ottawa, Canada (2000).