Eastern Illinois University
The Keep

Faculty Research & Creative Activity | Technology, School of

2013

Stochastic Performance Analysis of Distributed Activities

Toqeer Israr
Eastern Illinois University

Gregor v. Bochmann
University of Ottawa

Follow this and additional works at: http://thekeep.eiu.edu/tech_fac

Recommended Citation
Israr, Toqeer and Bochmann, Gregor v., "Stochastic Performance Analysis of Distributed Activities" (2013). *Faculty Research & Creative Activity*. 44.
http://thekeep.eiu.edu/tech_fac/44

This Conference Proceeding is brought to you for free and open access by the Technology, School of at The Keep. It has been accepted for inclusion in Faculty Research & Creative Activity by an authorized administrator of The Keep. For more information, please contact tabruns@eiu.edu.

Eastern Illinois University
The Keep

Faculty Research & Creative Activity | Technology, School of

January 2013

Stochastic Performance Analysis of Distributed Activities

Toqeer A. Israr
Eastern Illinois University, taisrar@eiu.edu

Follow this and additional works at: http://thekeep.eiu.edu/tech_fac

 Part of the Technology and Innovation Commons

Recommended Citation
Israr, Toqeer A., "Stochastic Performance Analysis of Distributed Activities" (2013). *Faculty Research & Creative Activity*. 23.
http://thekeep.eiu.edu/tech_fac/23

This Article is brought to you for free and open access by the Technology, School of at The Keep. It has been accepted for inclusion in Faculty Research & Creative Activity by an authorized administrator of The Keep. For more information, please contact tabruns@eiu.edu.

Stochastic Performance Analysis of Distributed Activities

Toqeer Israr, Gregor v. Bochmann

Department of Electrical Engineering and Computer Science
University of Ottawa,
800 King Edward Ave. Ottawa Ontario K1N 6N5 Canada
{tisra051, bochmann}@eecs.uottawa.ca

Abstract. This paper analyzes stochastic performance of a distributed global activity, composed of sub-activities sequenced serially, probabilistically, or concurrently. We provide general formulas with which we calculate the performance of a composite activity based on the performance of the constituent sub-activities and the control structure. To do this, we model each sub-activity as a Partially Ordered Specification (POS), where each sub-activity is characterized by independent input events, dependent output events and the stochastic minimum delays between these events. This technique allows two or more sub-activities to be combined hierarchically. Proofs of correctness for these formulas are given and a simple example is discussed throughout the paper.

Keywords: software performance, stochastic, modeling, partial order, collaborations, UML Activity Diagrams, distributed services, web services

1 Introduction

Many workflows consist of distributed systems, with multiple communicating components running on different processors or in different processes. For example, a *login* workflow may start with a request from a Web Client that invokes a process in a Retailer Server, which in turn makes calls to some "third party" Servers – components involved would be the Client, Retailer Server and the "third party" Servers.

A number of models have been employed during the system design and development process of such systems. These modeling paradigms include UML Activity Diagram [5], UML Activities [5], Use Case Maps, the Process Definition Language, Business Process Modeling Language, Business Process Execution Language, Web Services Choreography Description Language, and Petri Nets. Most, if not all, of the mentioned notations can potentially be decomposed in sub-activities and further sub-sub-activities. These notations assume a single system component to be allocated to a basic activity in the decomposition. However, this does not hold true anymore as even the most basic activity may involve several components. Clearly, there is a need to model when a component starts and ends its execution in a given workflow.

With the interaction of multiple components in a given workflow (activity), it is imperative sometimes to be able to calculate the performance of the components i.e. the delay from the beginning to the end of a component's participation.

To satisfy these needs to model activities, Bochmann et al. in [1] and [2] have proposed a new modeling paradigm based on UML Activities and their orderings whilst we represented this model in [3] as a partially ordered set of inputs and outputs, called Partially Ordered Specification.

While the aforementioned paradigms model the activities and analyze the correctness of the required communication protocols in [1] and [2], we analyzed the performance of such activities in [3]. Using partial orders, we introduced a Partially Ordered Specification (POS) to model the temporal relationships among the sub-activities within a given activity. Inspired by Performance Evaluation and Review Technique (PERT), with POS, we calculate the fixed completion time for each component (actor) of a global activity based on fixed performance characteristics of the sub-activities.

In this paper, we extend this work by assuming that delays are distributions, and analyze the performance of a global scenario based on the stochastic performance properties of the constituent sub-activities.

We start off by reviewing the composition rules for stochastic distributions in Section 2. In Section 3, we discuss the modeling paradigm based on activities as well as Partial Order Systems. We also describe the rules of strong and weak sequencing as well as provide performance analysis for fixed delays from [3]. In Section 4, we propose formulas and provide proofs for calculating the performance of composite activities with distributions.

2 Composition of Distributions

Graph-based models are common modeling paradigms to represent system behaviours as UML Activities. An Activity is comprised of actions (nodes) and sequencing operators (edges) such as sequence, alternative, concurrency, and loops to define the relationship between these actions, as illustrated in Fig 1. These activities may have constant or distribution time delays.

We analyzed in [3] the performance of a global activity, based on fixed time delays of constituent sub-activities. However, when activity delays are statistically varying and characterized by a time distribution, the analysis of a graph model can be quiet challenging. If an event-precedence graph with random activity times is in series/parallel/alternative in nature, the distribution function of the completion of the graph can be calculated by combining the distribution functions of the individual activities using multiplication and convolution [6]. However, the following initial assumptions are made:

- Parallel activities do not need to have identical distributions.
- Infinite resources are available for the activities and hence there are no resource contention issues.

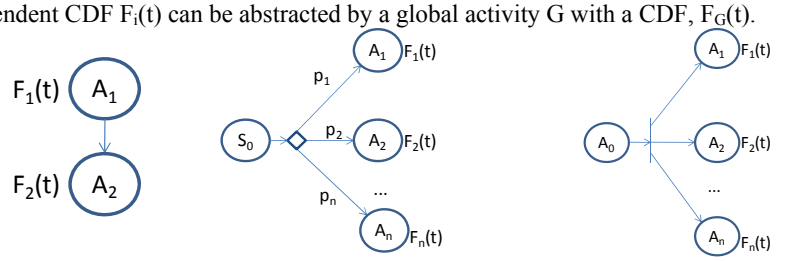
We assume the duration of an activity, A_i, has an associated delay distribution (probability density function) of $f_i(x)$. The cumulative distributive function (CDF) for the delay of activity A_i is then defined by: $F_i(t) = \int_{-\infty}^{t} f_i(x)\, dx$

The function $F_i(t)$, associated with each activity A_i, represents the probability that that activity A_i finishes by time t. While each sub-activity is assigned a cumulative

distribution function (CDF) for the completion time of the activity it models, these sub-activities may be composed to form a composite activity with a resultant CDF for the completion time of the entire set of considered activities.

We consider activities composed with the following 3 operators: series, alternative and concurrency. We assume all activities start with a single activity, called an initial activity. Activities are in series when a single activity succeeds the initial activity such as shown in Figure 1a. Alternate activities may exist when a single activity may execute amongst multiple activities such as shown in Figure 1b. Concurrent activities exist when multiple activities may execute simultaneously succeeding the initial activity.

Various compositions of these sub-activities $A_1 ... A_n$, each activity A_i with independent CDF $F_i(t)$ can be abstracted by a global activity G with a CDF, $F_G(t)$.



a) series b) alternative c) concurrency

Fig. 1. Various Operators

2.1 Series

If any two sub-activities, A_j and A_k, with CDF of $F_j(t)$ and $F_k(t)$ are combined in series such as shown in Figure 1a, the CDF of the combined activities is [6]:

$$F_{SUM}(t) = F_j(t) \otimes F_k(t) \quad (1)$$

where $\otimes$ represents convolution, defined as:

$$F_j(t) \otimes F_k(t) = \int_0^t F_k(t - x)\, dF_j(x) \quad (2)$$

2.2 Alternatives

If there is a probability p_i for the execution of activity A_i, such as in Figure 1b, the model represents a scenario with multiple possible paths. Then the distribution for the global activity G is [6]:

$$F_G(t) = \sum_{i=1}^{n} p_i(t)\, F_i(t) \quad (3)$$

2.3 Concurrency

If the sub-activities in the global activity G execute concurrently, such as in Figure 1c, then the following two cases can be considered.

Suppose there is a parallel search for an item in a distributed database, where the earliest concurrent search to finish, will terminate the overall search. We would be

interested in the time it would take for the earliest search to finish. A scenario with the earliest or "minimum" CDF of a global activity G composed of parallel sub-activities A_i, can be modeled by [6]: $${}_{\min}F_G(t) = 1 - \prod_{i=1}^{n}(1 - F_i(t)) \quad (4)$$

Now suppose there are parallel sub-activities, A_i, composing the global activity G again, but the completion of this global activity G requires all of the sub-activities to complete. This can be accomplished by calculating the delay of the activity with the maximum time delay by [6]: $${}_{\max}F_G(t) = \prod_{i=1}^{n} F_i(t) \quad (5)$$

Note: For the activity distributions in (4) and (5), the $F_i(t)$ need not to be the same for different *i*.

3 Modeling Distributed Activities

Based on [2], we introduced in [3] a Partially Ordered Specification (POS), which allows modeling of a UML activity with a partial ordered set of input and output events, as shown in Figure 2. This modeling paradigm shows the dependencies between various events and is used in analyzing performance of activities with various operators.

For a given activity, we suggested the input and the output of each involved role to be modeled by a starting and an ending event [3]. These events, belonging to various components, form a partially ordered set, where a causal relationship may exists between some of these events, shown by arcs "→" in the figure. The ending events are not ordered relative to one another, but each ending event has a dependency on its corresponding starting event (local sequencing). An initiating event, belonging to an initiating role (represented by dark circles "●"), is a specialized starting event in this partially ordered set of events, for which there are no other events in that set which precedes that event. Similar holds for the terminating events corresponding to terminating roles.

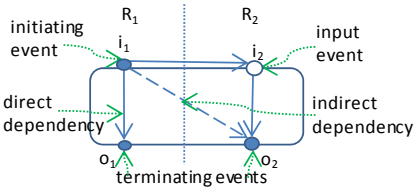


Figure 2 – Activity represent as a POS

Figure 2 illustrates an activity, with input events i_1 and i_2 and output events o_1 and o_2. As i_1 and o_1 are input and output events of the same role R_1, o_1 must occur after i_1 due to local sequencing. Furthermore, since i_1 is the only initiating event, all events in the activity, including, i_2, must occur after i_1. Due to the relationship $i_1 \rightarrow i_2$ and $i_2 \rightarrow o_2$, there is an indirect dependency from i_1 to o_2, shown by the dashed arrow "-->." Terminating events o_1 and o_2 are not ordered and may occur in parallel.

3.1 General Formulas for Standard Sequencing Operators with Fixed Delays

We introduced Nominal Execution Time Delay (NETD) written as Δ^{ix}_{om}, between input i_x and output o_m, where NETD is the the delay between the time instance of the occurrence of input event i_x and the occurrence of a dependent output event o_m, provided all the other events on which o_m depends have occurred long time ago [3].

Based on NETD, we derived the following general performance formula which can then be applied to sequencing operators of sub-activities to yield the performance metrics of a global activity:

$$t_{om} = \max_x (t_{ix} + \Delta^{ix}_{om}) \quad (6)$$

where t_{om} is the time of the output event o_m, t_{ix} is the time of the input event i_x on which o_m depends, and Δ^{ix}_{om} is the NETD from input event i_x to the output event o_m.

We assumed shared resources are not involved in these activities. Furthermore, we assumed that there is a dependency, and hence a delay, from each input to each output of an activity.

An activity may be comprised of sub-activities sequenced with strong or weak sequencing, parallel operators, alternatives and interruptions. In this paper, we limit our scope to strong and weak sequencing.

Strong sequencing, sometimes called global sequencing, between two activities A1 and A2 means that all sub-activities of A1 must be completed before any sub-activity of A2 may start. In contrast, weak sequencing between A1 and A2 (only) means that each system component locally applies sequencing to the local sub-activities of A1 and A2, that is, a component may start with sub-activities that belong to A2 as soon as it has completed all its local sub-activities that are part of A1. Strong sequencing implies weak sequencing, but not inversely. In particular, if a component is not involved in A1, it may start with sub-activities of A2 even before A1 begins its execution.

A strong sequence is modeled in Figure 3.0 between two sub-activities, A and B, and as discussed, all the initiating events in activity B may occur, only if all the terminating events of the previous activity, activity A, have occurred. This is represented by a Final Action event - when all the terminating events have occurred, denoted by AO_F (Final Output of sub-activity *A*). The time of all the initiating events for the next activity, activity B, is the time of the Final Action event of the previous activity, activity A.

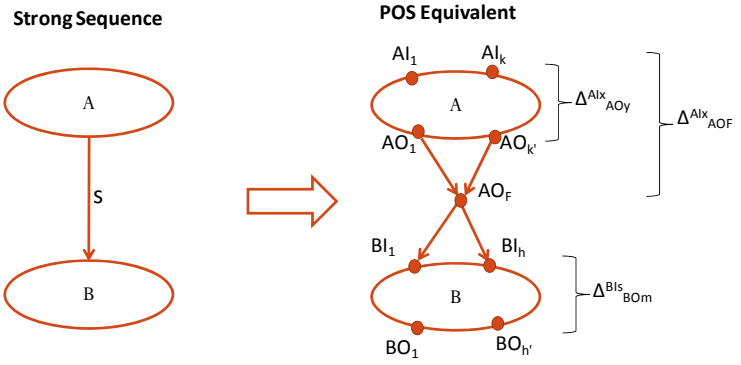


Fig. 2. Strong Sequencing

We also proposed and proved for two sub-activities, sub-activity A with k inputs and k' outputs and sub-activity B with h inputs and h' outputs, that are strongly sequenced, with known NETD for each sub-activity (Δ^{AIx}_{AOy} and Δ^{BIs}_{BOm}), that the NETD for the composite activity (Δ^{AIx}_{BOm}) is given by the formula:

$$\Delta^{AIx}{}_{BOm} = \max_{y=1..k'}(\Delta^{AIx}{}_{AOy}) + \max_{s=1..h}(\Delta^{BIs}{}_{BOm}) \quad (7)$$

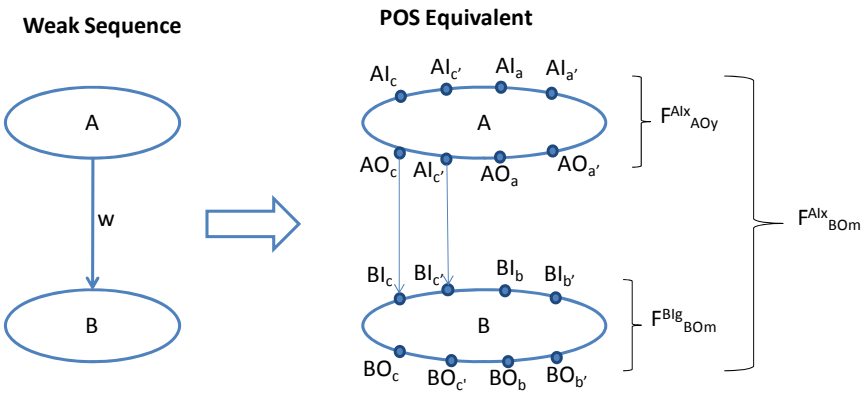


Fig. 3. Weak Sequencing

Similarly for two activities A and B weakly sequenced which have roles c..c' common to both A and B, it was shown that the NETD for the composition is:

$$\Delta^{AIx}{}_{BOm} = \max_{s=c..c'} (\Delta^{AIx}{}_{AOs} + \Delta^{Bis}{}_{BOm}) \quad (8)$$

4 General Formulas for Standard Sequencing Operators with Delay Distributions

So far we somewhat summarized the modeling and performance analysis of composite activities as described in [3]. The remainder of this paper is new material. In the following, we analyze the performance of activities composed of strong and weak sequences where the NETDs of activities are characterized by time distributions.

4.1 Strong Sequence

Proposition:
If two sub-activities, sub-activity A with k inputs and k' outputs and sub-activity B with h inputs and h' outputs, are strongly sequenced and the NETD for each sub-activity, $\Delta^{Aix}{}_{AOy}$ and $\Delta^{BIs}{}_{BOm}$, are known and characterized by cumulative distributive functions (CDF) $F^{Aix}{}_{AOy}(t)$ and $F^{BIs}{}_{BOm}(t)$, respectively, then the CDF, $F^{AIx}{}_{BOm}(t)$, of the NETD for the composite activity is:

$$F^{AIx}{}_{BOm}(t) = \prod_{y=1}^{k'} F^{Aix}{}_{AOy}(t) \otimes \prod_{s=1}^{h} F^{BIs}{}_{BOm}(t) \quad (9)$$

Proof:
Figure 2 shows strong sequencing between two activities A and B and its resultant

POS. We assume $F^{AIx}_{AOy}(t)$ and $F^{BIs}_{BOm}(t)$ are known, either given or measured using the testing methodology described in [3].

The delay from the Final Action event (AOF) relative to the input event x of A, is the maximum of all the paths' delays from event x to event AOF:

$$\Delta^{AIx}_{AOF} = \max_{y=1..k'}(\Delta^{AIx}_{AOy}) \quad (10)$$

Since the maximum CDF is required of all the delays involved, we use (5) and obtain:

$$F^{AIx}_{AOF}(t) = \prod_{y=1}^{k'} F^{Aix}_{AOy}(t) \quad (11)$$

The NETD for activity B for an input s to an output m is Δ^{BIs}_{BOm}. To calculate the maximum delay to produce the output m, the maximum is taken over all the given inputs of activity B:

$$\Delta^{BIj}_{BOm} = \max_{s=1..h}(\Delta^{BIs}_{BOm}) \quad (12)$$

Using (5), this formulas leads to:

$$F^{BIj}_{BOm}(t) = \prod_{s=1}^{h} F^{BIs}_{BOm}(t) \quad (13)$$

The delays for activity A and B are represented by (11) and (13), respectively. Since activity A and activity B are in sequence, using (1), the total time delay F^{AIx}_{BOM} is the convolution of $F^{AIx}_{AOF}(t)$ and $F^{BIj}_{BOm}(t)$:

$$F^{AIx}_{BOM}(t) = F^{AIx}_{AOF}(t) \otimes F^{BIj}_{BOm}(t) \quad (14)$$

$$= \prod_{y=1}^{k'} F^{Aix}_{AOy}(t) \otimes \prod_{s=1}^{h} F^{BIs}_{BOm}(t) \quad (15)$$

4.2 Weak Sequence

Weak sequencing between two activities A and B is illustrated in Figure 3. It is assumed that roles c…c' are participating in both activities A and B, while roles a…a' participate only in activity A and not in B and roles b…b' participate only in activity B and not in A.

To calculate the NETD between any two events, we assume all the remaining input events have occurred long time ago and any executions precipitated by these input events would have also completed long time ago as discussed in the testing methodology of [3]. Roles a…a' are involved only in activity A and no dependency exist from activity B to the output of roles a…a'. Hence, the NETD for roles a…a' is that of activity A, F^{AIx}_{AOy}. Similar is true for activity B.

We are interested in the NETD between the output events of activity B relative to the input events of activity A, given the NETDs are delay distributions.

Proposition:

If two activities, A and B, are weakly sequenced then the NETD between the output events of activity B relative to the input events of activity A is:

$$F^{AIx}_{BOm}(t) = \prod_{s=c}^{c'}(F^{AIx}_{AOs}(t) \otimes F^{Bis}_{BOm}(t)) \quad (16)$$

Proof:

The NETD of activity A and B is Δ^{AIx}_{AOy} and Δ^{Big}_{BOm}, respectively, which have the CDFs $F^{AIx}_{AOy}(t)$ and $F^{BIg}_{BOm}(t)$, respectively.

From the right side of Figure 3, it's evident that the NETD of the composed activity is the maximum of the NETD of both activities A and B in series with the common role *s*, i.e. *s* being the ending role of activity A and also the starting role of activity B.

Activity A, with input event x in series with activity B with output event m, is represented by $\Delta^{AIx}{}_{BOm}$ with the CDF $F^{AIx}{}_{BOm}(t)$. As two activities are in series, equation (1) can be used to calculate the NETD for a single *s*:

$$F_{single}{}^{AIx}{}_{BOm}(t) = F^{AIx}{}_{AOs}(t) \otimes F^{Bis}{}_{BOm}(t) \quad (17)$$

And to take the maximum over the inputs s=c…c', we obtain:

$$F^{AIx}{}_{BOm}(t) = \prod_{s=c}^{c'} (F^{AIx}{}_{AOs}(t) \otimes F^{Bis}{}_{BOm}(t)) \quad (18)$$

5 Conclusion

In this paper, we have considered the delay distributions of composite activities sequenced with strong and weak sequencing. Similar to analysis done in [4] for fixed delays, we plan to consider other composition operators such as parallel, alternatives, loops, etc for delay distributions. We have implemented a tool that takes as input an Activity Diagram including sub-activities with defined performance characteristics and provides as output the NETDs of the global collaboration for fixed delays. We plan on extending this tool to support delay distributions.

We believe that this approach to performance modeling of distributed systems is useful in many fields of application, including distributed work flow management systems, service composition for communication services, e-commerce applications, and Web Services.

References

1. Bochmann, G.V. Deriving component designs from global requirements, in: Proceedings on International Workshop on Model Based Architecting and Construction of Embedded Systems (ACES), Toulouse, 2008, pp 55-69
2. Castejón, H.N, Bræk, R., and Bochmann, G. v., "Realizability of Activity-based Service Specifications". Journal of Software and Systems Modeling(to be published), 2011
3. Israr, Bochmann, Performance Modeling of Distributed Activity Services, Proceedings of the 2nd ACM/SPEC International Conference on Performance Engineering, Karlsruhe, Germany, 2011, pp 475-480
4. Israr, Bochmann, Performance Modeling of Distributed Collaboration Services with Independent Inputs/Outputs, Proceedings of 5th International Workshop on Non-functional Properties in Modeling: Analysis, Languages, Processes, Miami, USA, 2013
5. OMG, Unified Modeling Language (UML), Version 2.1.1, February 2007
6. R.A. Sahner and K.S. Trivedi, Performance and reliability analysis using directed acyclic graphs. *IEEE Trans. Software Eng.*, (1987), pp. 1105–1114.