ifip

IFIP
WCC 2008
WORLD COMPUTER CONGRESS
Milano

AICA
Associazione Italiana per l'Informatica ed il Calcolo Automatico

20TH WORLD COMPUTER CONGRESS

MIC - fieramilanocity

September 7-10, 2008

Auctions in Cognitive Networks

Erol Gelenbe

The Challenges of Open-World Software

Carlo Ghezzi

Plenary keynotes

6147

in cooperation with:

INTERNATIONAL TECHNOLOGY GROUP

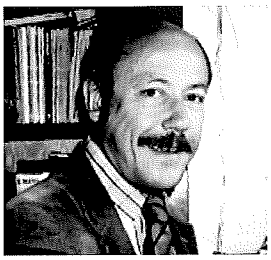
Main Partner

TELECOM ITALIA

Erol Gelenbe

Affiliation: Professor in the Dennis Gabor Chair, Electrical and Electronic Engineering Department, Imperial College London

Prof. Erol Gelenbe of Imperial College is a native of Turkey and graduated from Ankara Koleji and the Middle East Technical University in Ankara. He co-founded IFIP Working Group 7.3 which he chaired for two terms. He was elected a Fellow of the ACM and of IEEE, and was awarded the IFIP Silver Core in 1980. In the justification for his recent reception of the ACM SIGMETRICS Achievement Award 2008, it is said that he is "the single individual who, over a span of 30 years, has made the greatest overall contribution to the field of Computer System and Network Performance Evaluation through original research, mentoring and doctoral training, creation and direction of world class research groups, wide ranging international collaboration, and professional service" and that he "has made decisive contributions to product form networks by inventing G-networks (Gelenbe-Networks) with totally new types of <<negative customers, triggers, and resets>>, which are characterised by non-linear traffic equations. He has made seminal contributions to random access communications, the optimisation of reliability in database systems, the design of adaptive QoS-aware packet networks, diffusion models in performance analysis, and the performance of link control protocols." Prof. Gelenbe has authored four books in English and French, two of which have been published in Japanese and Korean. He has graduated 58 PhD students and published some 120 journal articles. His honours include: Commander of Merit of the Republic of Italy, Grand Officer of the Star of Italy, Officer Order of Merit of France and Chevalier des Palmes Académiques, and "honoris causa" doctorates from the University of Rome Tor Vergata (1996), Bogaziçi University, Istanbul (2004), and the University of Liège, Belgium. He is a Member of the French National Academy of Engineering (Académie des Technologies), the Turkish Academy of Sciences, and the Academia Europaea. He received the Science Award (1994) of the Parlar Foundation in Turkey, and was the first computer scientist to be awarded the Grand Prix France Telecom (1996) of the French Academy of Sciences. With very broad research interests, his recent publications include path finding algorithms in noisy and uncertain conditions, networked auctions, the use of neural networks to control routing in computer networks, as well as theoretical biology and theoretical chemistry including neural networks and gene regulatory networks. His work is funded by industry (GD, BAE Systems and QinetiQ), and research agencies including EPSRC, MoD, DoD, EU FP6 and FP7.



Carlo Ghezzi

Affiliation: Professor of Software Engineering in the Department of Electronics and Information of Politecnico di Milano

Carlo Ghezzi is the Rector's Delegate for research. In the past, he has been a member of the Academic Senate and a Member of the Board of Governors of Politecnico. He also held positions as Department Chair, Head of the PhD Program and Head of "consiglio di corso di laurea". He has also been affiliated with: Università di Padova, University of California at Los Angeles, University of North Carolina at Chapel Hill, University of California at Santa Barbara (USA), Escuela Superior Latino-Americana de Informatica (Argentina), Technical University of Vienna and University of Klagenfurt (Austria), University of Lugano (Switzerland).

He is a Fellow of the ACM and Fellow of the IEEE. He was awarded the ACM SIGSOFT Distinguished Service Award. He is a member of the Academy of Sciences (Istituto Lombardo). He has been a member of several governmental committees and was the Italian representative in the EU Information Technology Committee (Esprit Programme) during the 4th Framework Programme. He has been on the board of several international research projects and institutions in Europe, Japan, and the USA.

He is a regular member of the program committee of flagship conferences of the software engineering field, such as the International Conference on Software Engineering and Foundations of Software Engineering/ European Software Engineering Conference, for which he also served as Program and General Chair. He has been General Chair of the International Conference on Service Oriented Computing (ICSOC 2006, Chicago, USA). He has been the Editor in Chief of the ACM Transactions on Software Engineering and Methodology. He is currently an Associate Editor of IEEE Transactions on Software Engineering, Associate Editor of Science of Computer Programming (Elsevier), Service Oriented Computing and Applications (Springer), and Software Process Improvement and Practice (J. Wiley and Sons).

Carlo Ghezzi's research has been focusing on different aspects of software engineering and programming languages. Currently, he is active in the area of software architectures, especially evolvable and distributed software architectures for ubiquitous and pervasive computer applications. He co-authored over 150 papers, almost all of which are published internationally, and 8 books. He coordinated several national and international (EU funded) research projects.

Auctions in Cognitive Networks

Erol Gelenbe

Abstract The world's economy, including financial transactions, the selection and purchasing of goods, their transportation, logistics and sales, as well as design and production, are heavily dependent on the Internet. Within such an economy, there is a need for formal approaches such as auctions to automatically offer and accept bids for purchases and sales. Furthermore, the networks that are used for such systems will have to integrate the needs of networked economic applications in order to offer them assured adequate and automated service. This paper surveys some of our recent work in networked auctions and "smart" cognitive networks, which will unite to create an effective electronic economy.

1 Auctions and Cognitive Networks

Auctions are important economic activity model [10] since they allow the bidding and selling process to be formally defined and run in a timely fashion. Internet based auctions are increasingly popular and in the future this trend will be prevalent with networks of virtual buyer and seller agents conducting automated transactions across the world, linking different markets and commodities in complex interconnected economies [12, 14, 17]. Auctions are also used for decision making in resource allocation, and more generally they are an example of Computational Mechanisms [15] which are used for the coordination of systems of agents. At the other end, we can consider computer networks such as the Internet to be collections of "users and services" which interact with each other in a market-like framework [6].

When economic transactions are run in a network environment, failures in the network, packet losses, and delays in conveying information, become major hand-

Erol Gelenbe
Intelligent Systems and Neworks Group, Department of Electrical & Electronic Engineering, Imperial College, London SW7 2BT, UK e-mail: e.gelenbe@imperial.ac.uk

icaps for the economic agents. The effect of denial of service attacks can be even more dramatic in such a context. Thus the notion of network quality of service (QoS) has a direct impact on the success or failure of the economic agents using the network. Hence there is a direct relation between the "cognitive packet network" techniques [1, 2, 3] that have been invented to support QoS in networks, and the specific needs of a market economy. A network that is aware of the needs of the application in terms of economic transactions, as shown in Figures 1 and 2 can therefore greatly contribute to the resulting efficiency and economic well being of the market.

In [7, 8], we consider automated bidders and sellers which interact through a smart or cognitive network [4, 5]. We also consider networked auctions where bidders can circulate among a set of interconnected auctions. Human or automated buyers and sellers are rational agents pursuing possibly self-interested agendas. Thus a seller's objective may be to maximise their income. Similarly, buyers or bidders may wish to drive away the competition (for instance by moving the price up to a level where competitors shy away), and they would try to purchase a good at the lowest possible price. Thus "rationality" will also carry over to automated buyers and sellers acting under programmed algorithmic schemes. The frequency with which

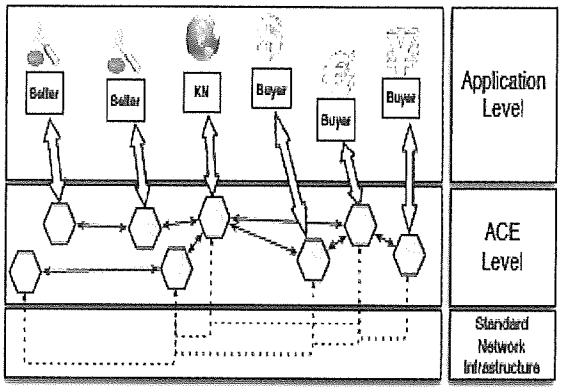


Fig. 1 An aucton system is managed by agents (ACE's) and resides on a network infrastructure

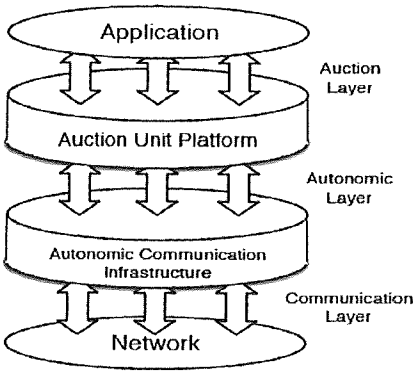


Fig. 2 The autonomic "cognitive" network manages access to the packet network so as to provide the best QoS to the auction system

bidders make their bids reflects how enthusiastic they are about purchasing the good. The"value" or valuation V attributed by buyers or sellers to a good also representats their rationality. We also include the time it takes the seller to take a decision to sell after receiving a bid as yet another element of the seller's rationality.

2 Auction model

We first analyse an auction in which bidders increase their offers in unit increments. We assume that bids arrive to the seller one at a time, according to a Poisson process of rate λ so that the average time that will elapse between successive bids is λ^{-1}. If a bid is not accepted by the seller, then the next bid will increment the value of the offer by 1 as long as the value V of the good for the buyers has not been reached. However the buyers will stop bidding and incrementing the offer when the most recent bid has reached the value V that the bidders associate with it.

We assume that the seller does not know the value V that buyers associate with the good, or that it makes no use of historical data about it. In any case our assumption is that the seller just relies on its ongoing observation about whether a new bid arrives. Thus after each bid, the seller waits for some random "decision time" to determine whether to accept the offer. If a new bid arrives before that time expires, then the process is repeated for this new bid. However if a new bid does not arrive before this time expires, then the seller accepts the current bid. Of course, if the seller accepts the offer too quickly, then the price he will be getting may be low with respect to the price that he would have gotten had he been more patient. On the other hand, if the buyer waits a long time before accepting an offer he may ultimately get the highest possible price (which is V), but at the expense of wasting a lot of time. Note that if bidders are parsimonious and only bid up to some price $R = V - h$, where $h \geq 0$ and represents the savings that the buyers hope to make on the price they pay, then we are back to dealing with an equivalent model where instead of V we use the random variable R. Note here that both V and h may be a random variable.

We may also have a situation where a bidder places a "time-out" on his offer so that the bid is withdrawn if the seller does not accept it before the time-out expires; however this case is not considered here, nor do we study cases where the seller varies the minimum sale price as a means to improve its income or accelerate the sale process. On the other hand, we study how the seller should choose this "decision time" so as to maximise his expected income over many successive sales, assuming that the value V is a random variable. Thus we will examine the relation between the average value of this random "decision time" and the other parameters of this system assuming that similar sales take place repeatedly after a "rest period" that follows after each individual auction. Another case which follows closely from the analysis in this section is when we consider that multiple units of the good are being sold at each auction. If the number of units sold at each successive auction

are independent and identically distributed random variables, all the analysis we develop here follows through very directly to this case.

Assume that after each bid, the seller waits for an exponentially distributed time of parameter d to accept the bid. Since the exponential distribution has the memoryless property, with this assumption the potential buyers cannot use their ongoing observations of the seller to determine when it will decide to accept an offer, since the remaining decision time at any point in time has the same distribution as the initial decision time right after a bid is made. If a new bid arrives, then the process is repeated. Furthermore assume that after a bid is accepted, the auction has a random "rest time" of average value R, which is again modelled via an exponentially distributed time of rate $r = R^{-1}$.

The auction repeats itself an infinite number of times, and all random variables and processes that we discuss are assumed to be independent from each other in each of the successive auctions, although they will always have the same distribution. In particular, the value V of the good beyond sold will be the random variable V_i for the $i-th$ successive auction, $i = 1, 2, \ldots$, with common distribution function $p(v)$. Thus for any of the auctions in the sequence, we model this system as a continuous time Markov chain $\{X_t : t \geq 0\}$ with state space $\{0, 1, \ldots, v, A_1, \ldots A_V\}$, where V is itself a random variable (so that we are dealing with a doubly stochastic process).

Let $\{t_1, t_2, \ldots t_n, \ldots\}$ be the sequence of instants when the successive auctions begin. Then:

- $X_{t_n} = 0$ corresponds to the value of the state when the $n-th$ auction begins,
- The state $X_{t_n+t} = l$, for $t > 0$ and $1 \leq l \leq v$, corresponds to the case at time $t_n + t$ during the $n-th$ auction $(t_n + t < t_{n+1})$ where l bids have been received so that the offered price has reached the value l, while
- $X_t = A_l$, $t > 0$, is the state at time $t_n + t$ during the $n-th$ auction $(t_n + t < t_{n+1})$ where the bid has been accepted at the price l.

Clearly, the transition rate from state l to $l+1$, $1 \leq l \leq v-1$, is λ. The transition rate from state l for $1 \leq l \leq v$ to state A_l is d, and the transition rate from any state A_l to state 0 is r.

In this model we assume that the good will be sold in each successive auction, and the main question is then to evaluate the *income per unit time* that the auctions generate for the seller. $P_v(.)$ denotes the stationary probability distribution of the Markov chain with state space $\{0, 1, \ldots, v, A_1, \ldots, A_v\}$, where we assume that V has a fixed value v. We will call this the *reduced* Markov chain, and denote by $\pi(l, v)$ the probability that the price that is obtained is l given that the value of the good is v.

The average rest time after the bid is accepted is the time spent in any of the states A_l, $1 \leq l \leq v$; it is identical (and given by r^{-1}) for each auction. Thus $\pi(l, v)$ is the probability that the reduced Markov chain is in state A_l given that it is in one of the states $\{A_1, \ldots, A_v\}$, hence:

$$\pi(l, v) = \frac{P_v(A_l)}{\sum_{i=1}^{v} P_v(A_i)}, \; 1 \leq l \leq v. \tag{1}$$

Assuming that the value of the good is a constant v, we can write the equations satisfied by the stationary probabilities $P_v(.)$ as follows:

$$P_v(l)(\lambda+d) = \lambda P_v(l-1),\ 1 \le l \le v-1, \tag{2}$$

$$P_v(0)\lambda = r\sum_{l=1}^{v} P_v(A_l), \tag{3}$$

$$P_v(v)d = \lambda P_v(v-1),$$

$$P_v(A_l)r = dP_v(l),\ for\ 1 \le l \le v,$$

$$1 = P_v(0) + \sum_{l=1}^{v}[P_v(l)+P_v(A_l)].$$

If we write $\rho = \frac{\lambda}{\lambda+d}$, after some calculations these equations yield:

$$P_v(l) = \rho^l P_v(0),\ 1 \le l \le v-1, \tag{4}$$

$$P_v(v) = \frac{\lambda}{d}\rho^{v-1}P_v(0),$$

$$P_v(A_l)r = dP_v(l),\ 1 \le l \le v,$$

$$P_v(0) = \frac{rd}{r\lambda+\lambda d+rd}.$$

As a consequence we can see that

$$\sum_{l=1}^{v} P_v(l) = \frac{\lambda}{d}P_v(0), \tag{5}$$

resulting in:

$$\pi(l,v) = \frac{d}{\lambda}\rho^l,\ 1 \le l \le v-1, \tag{6}$$

$$\pi(v,v) = \rho^{v-1}. \tag{7}$$

Using the probabilities $\pi(l,v)$ we can compute the expected income from a single auction for a good whose value is v directly as:

$$I = \frac{1-E[\rho^V]}{1-\rho} \tag{8}$$

We can also readily obtain the average income per unit time Φ for the auction as a whole, under the assumption that the value of the good being sold is a random variable V with probability distribution $\{p(v)\}$. We then have:

$$\Phi = \frac{\lambda r(\lambda+d)}{\lambda r+\lambda d+rd}[1-E[\rho^V]], \tag{9}$$

where $E[x^V] = \sum_{v=0}^{\infty} p(v)x^v$ is the generating function associated with the probability distribution $\{p(v)\}$. When we compare the formulas (8) and (9) we clearly see that

if we use the decision rate d to maximise income or income-per-unit time will lead to different results. However if the auction is "very busy" selling all the time, i.e. $r \rightarrow \infty$, we see that their ratio only depends on λ and d so that the best choice of d will be the same for this case:

$$lim_{r \rightarrow \infty} \; [\frac{I}{\Phi}] = \frac{\lambda + d}{\lambda d} \tag{10}$$

Of course, in the case $r \rightarrow \infty$ there is no rest period, and the seller is always busy running an auction.

It is interesting to observe how Φ varies as a function of d, as shown in the numerical examples of Figure 3. We select the value V of the good to be uniformly distributed between 80 and 120, we set $r = 1$ to normalise the remaining parameters, and plot the income per unit time against the decision rate d for different values of the arrival rate of the bids λ. We see clearly that there is indeed an "optimum" value of d which differs for each set of parameters and which maximises the income from the auction. Thus the seller can adjust his/her "patience" so that the rate of income is maximised. We also see that faster decisions (greater d) need to be made when the rate at which bids are made is higher. The curves show that there is some optimum value of d that will maximise the seller's income. Furthermore, when λ is large we can afford to be less careful about waiting for some best value of the average decision time, and we can make decisions quite quickly (d large).

3 A Rational Economic Bidder with n other bidders

In the previous analysis we considered all bidders to be indistinguishable. Now our purpose is to better understand how one particular rational bidder should behave in the presence of n other bidders that it is competing with, and which it can observe. We call this designated economic agent the Rational Economic Bidder (REB). We first consider an auction in which the REB participates with n other bidders. We assume that a single good is being sold, and that it has a maximum valuation $v > 0$,

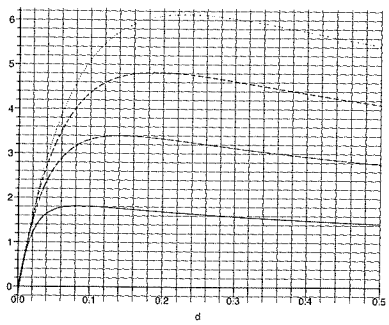


Fig. 3 Income per unit time versus the rate d at which decisions are made, for different values of the rate at which bids arrive ranging from 8 (top) to 2 (bottom). The value is uniformly distributed between 80 and 100. If the decision time is large (d less than 0.1), the income per unit time can drop significantly.

so that none of the bidders will bid beyond the sum of money v. Initially we will assume that v is a fixed and identical value for all the bidders, but it will be easy to generalise this to the case where v is a random variable so that the actual maximum value associated with a good may be known only in terms of a probability distribution. During the auction, each of the bidders may take some time to consider the current highest offer before deciding to place a new counter-offer. We assume that these thinking times are exponentially distributed random variables with parameters β and λ, for the REB and the other bidders, respectively, so that we may distinguish the behaviour of the REB from all the other bidders which have a common statistical behaviour. All the bidders can participate in submitting bids, except obviously for the bidder who owns the current highest offer. Furthermore we assume that all bids proceed with unit increments with respect to the previous bid, in order to minimally surpass previous highest bid. The system that we have described is modelled as a continuous time Markov chain $\{X_t : t \geq 0\}$ with

$$X_t \in Y = \{0, O(l), R(l), A(O,l), A(R,l) : 1 \leq l \leq v\} \tag{11}$$

Initially we have $X_0 = 0$ and the state values are described as follows for any t:

- $X_t = 0$, if no bid is placed at time t. Note that this may occur after any one of the instants $t_{i+1} = \inf\{t > t_i \textit{ and } t_{i+1} = 0\}$ when the seller accepts a bid, and the auction restarts. We set $t_0 = 0$.
- $X_t = O(l)$ where $0 < l \leq v$, if at time t the current value of the bid is l and the current bidder is not the REB, regardless of who placed the previous $l-1$ bids.
- $X_t = R(l)$ where $0 < l \leq v$, if at time t the current bidder is the REB and the value of his bid, i.e. the current highest bid, is l.
- $X_t = A(O,l)$, if at time t the auction has concluded with a sale at price l to one of the "other" n bidders, i.e. other than the REB, and the next auction has not yet restarted.
- $X_t = A(R,l)$, if at time t the auction has concluded with a sale at price l to the REB, and the next auction has not yet restarted.

Any bidder which is not the current highest bidder can place a bid at rate β and λ, respectively, for REB and the other bidders, as long as the current bid value has not attained v. When the valuation v has been attained, no further bids will be placed. Also, the transition rate that denotes the start of a new auction, from either state $A(O,l)$ or $A(R,l)$ to state 0 is r, and the transition rate (denoting the seller's decision to sell) from state $O(l)$ to $A(O,l)$ and from $R(l)$ to $A(R,l)$ is δ. Note that the seller cannot tell the difference between the REB and the other bidders, and the transition rates in this first model do not depend on the current value of the highest bid. For any state $x \in Y$, the stationary distribution is: $P(x) = \lim_{t\to\infty} P\{\ X_t = x\ \}$, and the balance equations satisfied by the stationary probabilities of the system are:

$$\begin{aligned} &P(O(1))((n-1)\lambda + \beta + \delta) = n\lambda P(0), \\ &P(O(l))((n-1)\lambda + \beta + \delta) = (n-1)\lambda P(O(l-1)) \\ &\quad + n\lambda P(R(l-1)), 2 \leq l \leq v-1 \end{aligned} \tag{12}$$

$$
\begin{aligned}
&P(O(v))\delta = (n-1)\lambda P(O(v-1)) \\
&\quad + n\lambda P(R(v-1)), \\
&P(A(O,l))r = \delta P(O(l)), \quad 1 \le l \le v \\
&P(R(1))(n\lambda+\delta) = \beta P(0), \\
&P(R(l))(n\lambda+\delta) = \beta P(O(l-1)), 2 \le l \le v-1 \\
&P(R(v))\delta = \beta P(O(v-1)), \\
&P(A(R,l))r = \delta P(R(l)), \quad 1 \le l \le v \\
&1 = P(0) + \sum_{U=0,R} \sum_{l=1}^{v} \left[P(U(l)) + P(A(U,l))\right]
\end{aligned}
$$

Solving these equations, we obtain the closed form expression for $H(l)$ where $1 \ge l \ge v-1$. If the value v is replaced by a random variable V with the same distribution function both for the REB and the other bidders, then the analysis follows directly from the previous discussion, except that all the previous quantities have to be computed as expectations with respect to the random variable V which has a general probability distribution $Prob[V=v] = p(v)$. Some measures of interest to the REB are:

- (a) whether the REB is actually able to purchase the item it is seeking,
- (b) how fast it can purchase the item,
- (c) whether it is able to minimize the cost of its purchase or equivalently how much it saves with respect to the maximum price that it is willing to pay, and
- (d) what is its savings per unit time with respect to the maximum price v that it might have paid.

The seller's interest is to maximise its income from a sale, or to maximise its income per unit time for a sequence of sales. Note that $P(0)$ is the ratio of the average time elapsing from when the auction starts until the first bid arrives, to the total average time τ an auction cycle lasts (including the "rest time" of average value r^{-1} after an auction ends). Since the system leaves state 0 only when the first bid in an auction is made, the average time spent in this state is simply the inverse of the rate at which the first bid is made, i.e. $[n\lambda+\beta]$, and

$$
\begin{aligned}
P(0) &= \frac{\textit{Average time in state } 0}{\tau} \qquad (13) \\
\tau &= \frac{P(0)^{-1}}{n\lambda+\beta} \\
&= \frac{r\delta + (r+\delta)(n\lambda+\beta)}{r\delta(n\lambda+\beta)}
\end{aligned}
$$

When a sale is made, the *expected income of the seller* is

$$I = \frac{\sum_{l=1}^{v} l[P(A(R,l)) + P(A(O,l))]}{\sum_{l=1}^{v} [P(A(R,l)) + P(A(O,l))]} \tag{14}$$

and the seller's income per unit time is

$$\iota = \frac{I}{\tau} \tag{15}$$

Finally, π the *probability that the REB is the bidder that makes the purchase* at an auction, rather than one of the other bidders, is given by:

$$\begin{aligned} \pi &= \frac{\sum_{l=1}^{v} P(A(R,l))}{\sum_{l=1}^{v} [P(A(R,l)) + P(A(O,l))]} \\ &= \left[\sum_{l=1}^{v} P(A(R,l))\right] \cdot \left[\frac{r\delta + (n\lambda + \beta)(r+\delta)}{\delta(n\lambda + \beta)}\right] \end{aligned} \tag{16}$$

Hence the average time ψ that the REB waits to win an auction is the inverse of its winning rate or

$$\psi = \frac{\tau}{\pi} = \frac{1}{r\sum_{l=1}^{v} P(A(R,l))} \tag{17}$$

Finally, another measure of interest is the average difference between the valuation v for the good, and the price at which the auction concludes, given that it wins, which we denote by:

$$\phi(v) = \frac{\sum_{l=1}^{v} (v-l)P(A(R,l))}{\sum_{l=1}^{v} P(A(R,l))} \tag{18}$$

Clearly, without reverting to deceit, all the REB can do is to adjust its bidding rate β to the situation it is observing, including the bid rate it observes concerning other bidders, so as to optimise the performance measures that it is selfishly interested in. Obviously, in order to maximise $\psi(v)$ it would suffice to take $\beta >> n\lambda$ so that the REB raises its bid to the value v very quickly so that it is always the winner, and $\pi(v)$ then tends to 1. However this means that the REB would be buying the good at its maximum price, rather than driving a good bargain. Thus a reasonable approach would be to choose a value of β which maximises the REB's return on the auction, such as $\gamma(v)$, the average savings per unit time that the REB makes with respect to the maximum price that it would pay, or specifically:

$$\gamma(v) = \frac{\phi(v)}{\psi} = r\sum_{l=1}^{v} (v-l)P(A(R,l)) \tag{19}$$

If v is replaced by the random variable V, the function of interest to the REB is:

$$\Gamma = E[\gamma(V)] = r \sum_{v=1}^{\infty} \sum_{l=1}^{v} (v-l)p(v)P(A(R,l)) \tag{20}$$

An interesting question arises if the REB adjusts its bidding rate β in a manner proprtional to the bidding rate of all other bidders. From the state equations (12) we see that if:

$$\mu = \frac{\beta}{n\lambda + \delta} = \frac{(n-1)\lambda}{(n-1)\lambda + \beta + \delta} \tag{21}$$

then the outcome of the auction for the REB will be equivalent to that for the other bidders taken together. In fact if n is large enough, this simplifies to:

$$\mu \approx \frac{1}{2}[\sqrt{1 + 4\frac{n\lambda}{n\lambda + \delta}} - 1] \tag{22}$$

In figures 4 to 7 we show various system measures as functions of varying λ and δ, when REB follows this policy in keeping up with the other bidders.

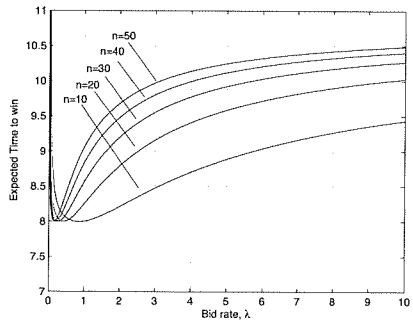


Fig. 4 Expected time to win for REB when keeping up with the other bidders for various λ and n. Other parameters: $\delta = 0.5$, $r = 1$.

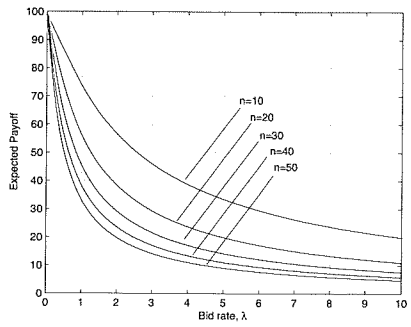


Fig. 5 Expected payoff for REB when keeping up with the other bidders for various λ and n. Other parameters: $\delta = 0.5$, $r = 1$.

4 Networked auctions

We now extend this work to bidders which seek to purchase goods in English auctions that are being conducted by different sellers. Although we imagine here that all these auctions are selling the same type of good, the analysis would be similar if the networked system includes auctions for different types of goods. Consider N auctions which are running concurrently, and let

$$n(t) = (n_1(t), \ldots, n_N(t)) \tag{23}$$

be the vector representing the number of potential bidders $n_i(t)$ at the ith auction at time $t \geq 0$, while the vector

$$k(t) = (k_1(t), \ldots, k_N(t)) \tag{24}$$

denotes the price currently attained by the most recent bids at the N auctions, and $k_i(t)$ is the price that the ith auctions have reached at time t, so that the state of the system as a whole is the pair of vectors $(n(t), k(t))$, which will have the following constraints:

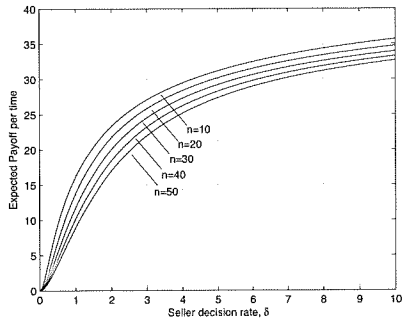


Fig. 6 Expected payoff per time for REB when keeping up with the other bidders for various δ and n. Other parameters: $\lambda = 1, r = 1$.

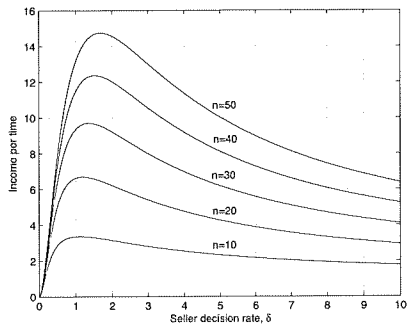


Fig. 7 Expected income per time for the seller when REB keeps up with the other bidders for various δ and n. Other parameters: $\lambda = 1$, $r = 1$.

- If $k_i(t) > 0$, this means that there is at least one current bid at auction i; as a consequence we must have $n_i(t) > 0$ as well since there must be at least one potential customer present.
- If $n_i(t) = 0$ then $k_i(t) = 0$, while if $k_i(t) = 0$ then we may have any value $n_i(t) \geq 0$.
- If $n_i(t) > 0$ it is possible that no bid has yet been received at auction i or that the most recent bid has been accepted and the successful bidder has returned to the set of other bids, so that we may have $k_i(t) = 0$.

We introduce the *Mobile Bidder Model (MBM)* where any bidder, except for the one who has made the most recent bid and is waiting for a response from the seller, is allowed to move at any time from one auction to another one. Of course, all other bidders are also allowed to make a bid. Bidders who leave one auction can either go to another one or they can depart from the auction system as a whole. In this model, the mobility of bidders will be represented by a Markov chain with transition probabilities $P(i,j)$ where $i \in \{1, .. N\}$ and $j \in \{1, .. N, D\}$, where D denotes the departure point of a bidder from the system, so that

$$\sum_{j=1}^{N} P(i,j) + P(i,D) = 1, \tag{25}$$

for any $i = 1, \ldots N$. Furthermore, bidders join auction i from the "outside world" according to a Poisson process.

The MBM is represented as follows, where we denote by y the number of bidders and by x the level attained by the bid:

- Bidders arrive from the "outside world" to the ith auction according to a Poisson process of rate γ_i.
- The rate of departure from auction i of a bidder, except for the one that has made the current highest bid, is $\mu_i \geq 0$ so that with probability $\mu_i \Delta t + o(\Delta t)$ any one bidder leaves the auction in the time interval $[t, t+\Delta t[$. If there is currently no outsanding bid, all bidders can depart. Thus the rate of departure is a function of (y,x) as follows:

$$\mu_i(y,x) = (y-1)\mu_i, \; x > 0, \tag{26}$$
$$\mu_i(y,0) = y\mu_i. \tag{27}$$

  A bidder departing from auction i will head to auction j, or to the outside of the network if $j = D$, with probability $P(i,j)$, where $j \in \{1, \ldots N, D\}$.
- The value of the good being sold at auction i is the random variable V_i which is independent of other quantities in the system, and its probability distribution is $p_i(v) = Prob[V_i = v]$, and let $\psi_{i,x} = Prob[x < V_i]$ with $\psi_{i,0} = 1$. New bids will stop being made when the most recent bid has attained the value of the good. Thus the rate at which bidders at auction i make bids is a function of (y,x) as follows:

$$\beta_i(y,x) = (y-1)\beta_i \psi_{i,x} \; x > 0, \tag{28}$$
$$\beta_i(y,0) = \beta_i y.$$

where $\beta_i > 0$. Again for any individual bidder we assume exponentially distributed bidding times, independent of others' bids and of the same bidder's previous bids.

- Finally, the rate at which a bid is accepted at auction i is δ_i, and the bid acceptance times are independent and identically distributed random variables. When a bid is accepted at auction i the successful bidder joins the set of bidders at the same auction, and auction i starts again with all the bidders. Notice that the most recently winning bidder's behaviour henceforth is the same as that of the other bidders.

With these assumptions, $\{(n(t),k(t)),t \geq 0\}$ is a continuous time, discrete state-space Markov chain. We denote its probability distribution by $p(n,k,t) = Prob[n(t) = n, k(t) = k]$ for some given initial condition at $t = 0$. Let us introduce the following notation. e_i will denote the $n-vector$ all of whose elements are zero, except for the ith element which is $+1$, and let $n_i^+ = n + e_i$, $n_i^- = n - e_i$ provided that $n_i > 0$, $n_{ij}^{+-} = n + e_i - e_j$ provided that $n_j > 0$. Similarly $k_i^+ = k + e_i$ and $k_i^- = k - e_i$ provided that $k_i > 0$. The Chapman-Kolmogorov (C-K) equations for the system are:

$$\frac{dp(n,k,t)}{dt} = \sum_{i=1}^{N} \{ \gamma_i p(n_i^-,k,t) 1[n_i > 0] \tag{29}$$

$$\begin{aligned}
&+ \beta_i(n_i - 1)\psi_i(k_i - 1)p(n,k_i^-,t)1[n_i > 0, k_i \geq 2] \\
&+ \beta_i \psi_{i,0} n_i p(n,k_i^-,t)1[k_i = 1] \\
&+ \mu_i P(i,D)p(n_i^+,k,t)(n_i 1[k_i > 0] + (n_i + 1)1[k_i = 0]) \\
&+ \sum_{j=1}^{N} \mu_i n_i P(i,j)p(n_{ij}^{+-},k,t)1[n_j > 0, k_i > 0] \\
&+ \sum_{j=1}^{N} \mu_i (n_i + 1)P(i,j)p(n_{ij}^{+-},k,t)1[k_i = 0] \\
&+ \sum_{x=1}^{\infty} \delta_i p(n,k + xe_i)1[k_i = 0]1[n_i > 0] \} \\
&- \sum_{i=1}^{N} \{\gamma_i + [(\delta_i + \mu_i(n_i - 1))1[n_i > 0, k_i > 0] \\
&\quad + \beta_i \psi_i(k_i)(n_i - 1)]1[n_i > 0, k_i > 0] \\
&\quad + (\beta_i n_i + \mu_i n_i)[n_i > 0, k_i = 0] \} p(n,k,t)
\end{aligned} \tag{30}$$

Note that the term beginning with δ_i in the 7-th line of the equation corresponds to the transitions which occur when bids are accepted, while the last three lines correspond to the rates of the events that do not take place.

Under the *Active Bidders Assumption (ABA)*, we suppose that that

$$p(n,k,t)1[k_i = 0] << p(n,t) \quad if\ n_i > 0, \tag{31}$$

i.e., for any (n,k) such that $n_i > 0$ and $k_i = 0$. We are thus assuming that the probability that there are *no bids* when there is at least *one bidder* at an auction is *very small* compared to the overall probability of the same state.

The stationary solution of equations for the MBM under the ABA are obtained by setting $\frac{dp(n,t)}{dt} = 0$ for all $n_i > 0$ in and is given by:

$$p(n) \approx \prod_{i=1}^{N} \frac{e^{-U_i}}{U_i} \frac{U_i^{n_i}}{(n_i - 1)!} \tag{32}$$

where $U_i = \frac{\Lambda_i}{\mu_i}$ and the Λ_i are the solution of the system of linear equations:

$$\Lambda_i = \gamma_i + \sum_{j=1}^{N} \Lambda_j P(j,i), \; i = 1, \; \ldots, N \tag{33}$$

Furthermore, in steady state the average number of bidders at the ith auction is given by $< n_i > \; \approx \; [1 + U_i]$.

4.1 The solution when bids and sales are very frequent

In this section we will consider the case when bidding and selling occurs more frequently than the bidders' moving between auction centres and into our out of the auction centre. Thus we will consider the case where for all $i = 1, \; \ldots, N$,

$$\Lambda_i, \; \mu_i \; << \; \delta_i, \; \beta_i \psi_{i,k_i} \tag{34}$$

which also implies that $\gamma_i << \delta_i, \; \beta_i \psi_{i,k_i}$. As an immediate consequence we have that the solution $p(k|n)$ for $n_i > 0$ and $k_i > 0$, $i = 1, \; \ldots N$ is given by:

$$p(k|n) \approx \prod_{i=1}^{N} p_i(k_i|n_i), \; where \tag{35}$$

$$p_i(k_i|n_i) = p_i(0|n_i) \prod_{l=1}^{k_i} [\frac{\beta_i \psi_{i,l-1}(n_i - 1)}{\gamma_i + \beta_i \psi_{i,l}(n_i - 1)} \tag{36}$$

and

$$p_i(0|n_i) = [1 + \sum_{k_i=1}^{\infty} \prod_{l=1}^{k_i} [\frac{\beta_i \psi_{i,l-1}(n_i - 1)}{\gamma_i + \beta_i \psi_{i,l}(n_i - 1)}]^{-1}. \tag{37}$$

where we have used the fact that the conditional probabilities sum to one. Finally, we can write the steady-state solution under the ABA when bids and sales are frequent using (36) as:

$$p(n,k) \approx p(k|n) p(n) \tag{38}$$

$$\approx \prod_{i=1}^{N} \frac{e^{-U_i}}{U_i} \frac{U_i^{n_i}}{(n_i - 1)!} p(0|n_i) \prod_{l=1}^{k_i} [\frac{\beta_i \psi_{i,l-1}(n_i - 1)}{\gamma_i + \beta_i \psi_{i,l}(n_i - 1)} \tag{39}$$

An important property of these approximate solutions is that they have "product form". This means that once we have solved the traffic equations (33) which allow us to compute the rate at which bidders arrive to each of the auctions, we can obtain the marginal probability distributions for any of the auction centres separately, while the overall joint probability distribution is obtained from the product of thes emarginal distributions.

5 Conclusions

The world's economy is increasingly automated and dependent on the Internet. Thus we will need to better understand how, on the one hand automated transactions can take place and what factors will affect the resulting levels of economic performance. On the other hand we will also need to understand how the Internet itself can be organised so as to offer the best service to economic transactions. This paper is an initial investigation into these issues.

We have considered specific economic transaction models as auctions with "automated" bidders and sellers, where bidders arrive to auctions according to some random process, and where the time it takes a seller to make a decision is also modelled as a random variable. We analyse such systems to obtain closed form expressions for their steady state behaviour as a function of different system paramaters. We consider metrics such as the sale price reached by a good in an auction, and the income per unit time that the seller obtains from repeated sales of the same good to different buyers. We also consider the impact of the time that it takes a seller to reach its decision on the metrics of interest. We have obtained analytical solutions for such systems, bringing into evidence the interplay between the critical parameters that can be modified by the buyers and sellers to their advantage.

Many interesting questions need to be considered in future work. The analysis we have presented leads to problems of optimisation and estimation. In the networked case, future work should study the mechanisms of price formation in such systems. Furthermore we need to better understand how network performance failures, such as packet delays, and delays in communicating decisions or information, as well as losses of information, and network attacks, can impact the performance of an automated and networked economy.

References

1. E. Gelenbe, R. Lent and Z. Xu (2001). Measurement and performance of a cognitive packet network. Computer Networks 37 (6): 691–701.
2. E. Gelenbe and R. Lent and A. Montuori and Z. Xu (2002). Towards networks with cognitive packets. Proceedings of the International Conference on Performance and QoS of Next Generation Networking, 3–17, Nagoya, Japan.

3. E. Gelenbe, R. Lent, A. Montuori and Z. Xu (2002). Cognitive packet networks: QoS and performance. Proceedings of the IEEE MASCOTS Conference, 3–12, Ft. Worth, USA.
4. E. Gelenbe, R. Lent and A. Nunez (2004). Self-aware networks and QoS. Proceedings of the IEEE, 92 (9): 1478–1489.
5. E. Gelenbe, M. Gellman, R. Lent, P. Liu and Pu Su (2004). Autonomous smart routing for network QoS. Proceedings of the First International Conference on Autonomic Computing, 232–239, New York, USA.
6. E. Gelenbe (2005). Users and Services in Intelligent Networks. Lecture Notes in Computer Vol. 3837: 30–45, Springer, Berlin & Heidelberg.
7. E. Gelenbe (2006). Analysis of automated auctions. Lecture Notes in Computer Science Vol. 2463: 1-12, Springer, Berlin & Heidelberg.
8. E. Gelenbe (2008). Analysis of single and networked auctions. Accepted for publication in *ACM Transactions on Internet Technology*.
9. E. Gelenbe and I. Mitrani (1980). Analysis and Synthesis of Computer Systems. Academic Press, New York and London.
10. P. R. Milgrom and R. Weber (1982). A theory of auctions and competitive bidding", *Econometrica*, 50: 1089-1122.
11. R.P. McAfee and J. McMillan (1987). Auctions and bidding", *J. Economic Literature*, 25: 699–738.
12. O. Shehory. (2001). Optimality and risk in purchase from multiple auctions", in M. Klusch and F. Zambonelli (Eds.), Lecture Notes in Artificial Intelligence, Vol. 2182: 142-153. Springer.
13. X. Guo (2002).An optimal strategy for sellers in an online auction. *ACM Trans. Internet Tech.* 2 (1): 1-13.
14. O. Shehory (2002). Optimal bidding in multiple concurrent auctions. *International Journal of Cooperative Information Systems*, Vol. 11(3-4): 315-327.
15. N.R. Dash, N.R. Jennings and D.C. Parkes (2003). Computational mechanism design: a call to arms. *IEEE Intelligent Systems*, 40-47, Nov.-Dec..
16. M. T. Hajiaghayi, R. Kleinberg, and D.C. Parkes (2004). Adaptive limited-supply online auctions. *Proc. 5th ACM Conference on Electronic Commerce*, ACM Press, 71-90, New York.
17. E. David, A. Rogers, J. Schiff, S. Kraus, and N. R. Jennings (2005). Optimal design of English auctions with discrete bid levels", *Proc. of 6th ACM Conference on Electronic Commerce (EC'05)* Vancouver, Canada, 98-107.
18. S. Fatima, M. Wooldridge and N. R. Jennings (2005). Sequential auctions for objects with common and private values. *Proc. 4th Int Joint Conf on Autonomous Agents and Multi-Agent Systems*, Utrecht, Netherlands, 635-642.

deep se dependable evolvable pervasive software engineering

The Challenges of Open-World Software

Carlo Ghezzi
DeepSE Group
Dipartimento di Elettronica e Informazione
Politecnico di Milano, Italy
carlo.ghezzi@polimi.it

Plan of the talk

- We are moving towards **open-world software**
- Briefly revisit software evolution through the **closed** to **open** world perspective
- Identify challenges and opportunities
- Current research contributions
- Future work

From "closed" to "open" world

- Progress in software methodologies and technologies can be understood through an evolution of the fundamental assumptions made on the relation between the system and the external world
  - ***closed world***
  - to ***semi-open world***
  - to ***open world***

The external world with which the system interacts is fixed and known

The structure of the solution is fixed, monolithic, and centralized

From "closed" to "open" world

- Progress in software methodologies and technologies can be understood through an evolution of the fundamental assumptions made on the relation between the system and the external world
  - ***closed world***
  - to ***semi-open world***
  - to ***open world***

Changes in the external world can occur, but can be anticipated

Changes in the architecture can be hidden behind stable interfaces

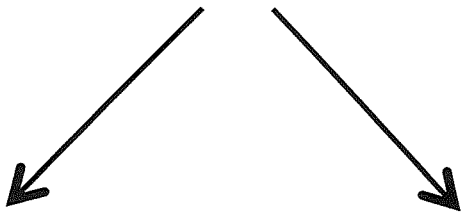
From "closed" to "open" world

- Progress in software methodologies and technologies can be understood through an evolution of the fundamental assumptions made on the relation between the system and the external world
  - ***closed world***
  - to ***semi-open world***
  - to ***open world***

Changes in the world are continuous and can occur as system is running
Run-time evolution of architecture

Closed world

- Assumptions
  - Boundary system/world fixed and immutable



- Architecture
  - Monolithic, centralized, static
- Process
  - Monolithic, Waterfall (phased/sequential)
  - Top-down, deductive from requirements

Semi-open world

- Assumptions
  - Changes in the (business/physical) world are inevitable, but they are slow and can be anticipated

- Architecture
  - Interface/implementation
  - Encapsulation
  - Object-orientation

- Process
  - Evolutionary, spiral
  - Multi-site developments (off-shoring)

Components and CBSE

Achievements

- Methodologies and tools for design-for-change
- Methodologies and tools for software re-engineering
- Interpretation models of how things get done via empirical studies
- Global software development
  - Decentralized developments
  - Centralized control
- Components
  - Multiple stakeholders

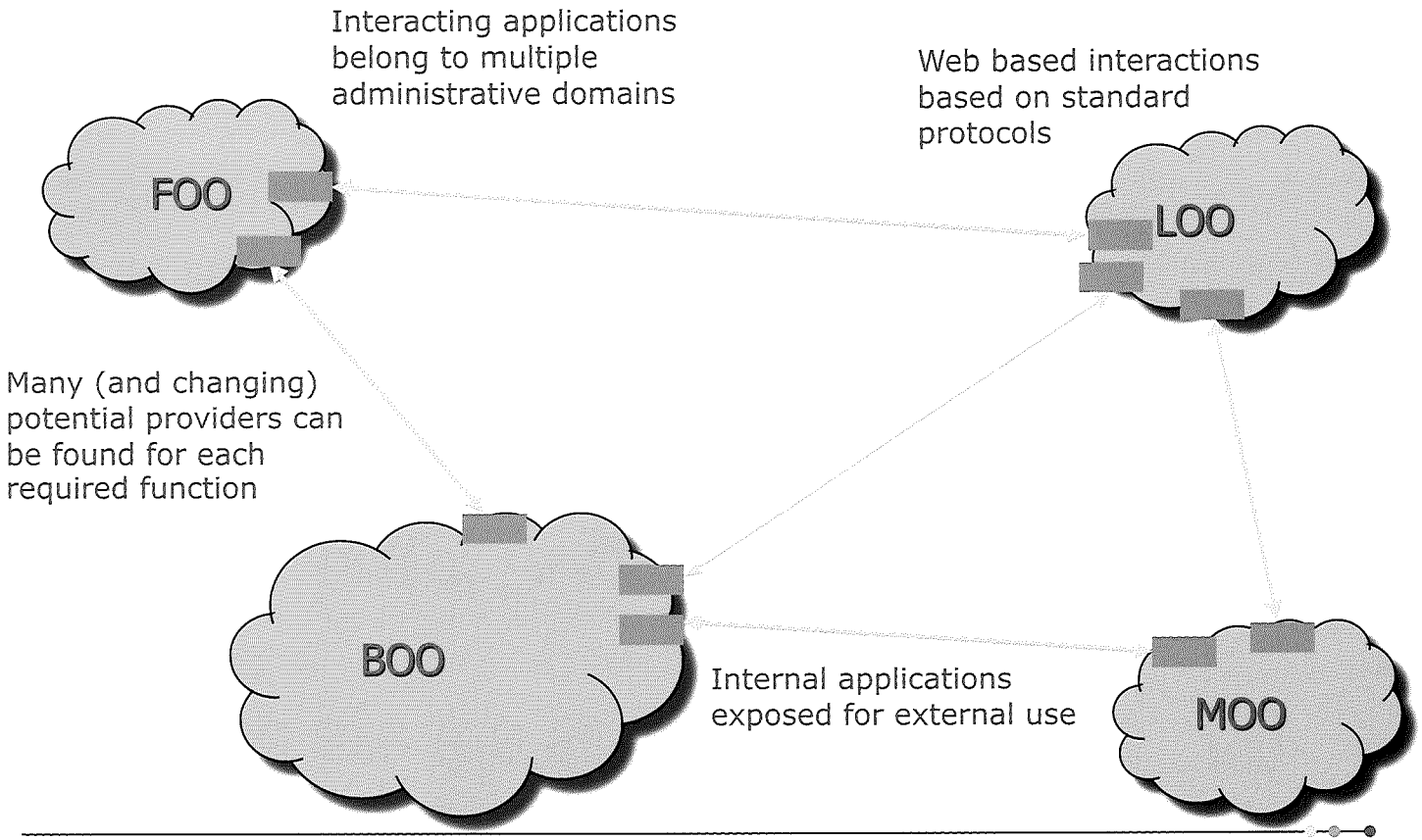
What is happening today?

- Unprecedented degrees of change
  - "Perpetual beta"
- Open world
  - Product
  - Process

Where are the sources of change? (1)

- Changes originate in the business world
  - Agile networked organizations
    - dynamic, goal-oriented, opportunistic federations
    - reacting with fast responses to rapidly changing requirements and changes in business world
  - Changes require dynamic adaptation in the software architecture

Networked organizations



Where are the sources of change? (2)

- The Internet of Things
  - Physical objects become part of the system
- Ubiquitous/ pervasive computing settings
  - Any***where***, any***time***, any***thing***
- Context awareness
- Ambient intelligence

Open world (1)

- In an open world requirements change continuously and unpredictably
- Functionalities to offer depend on context
- The system must adapt accordingly, as it runs and provides service
  - ***context-aware, situational computing***

Open world (2)

- From components to services
  - Both are developed by providers (no single authority is in charge of all parts)
    - Decentralized developments
  - Components deployed and run by the owner of the system
  - Services deployed and run by providers
  - Components selected/composed (bound) at design/ construction time
  - Services chosen and bound at run-time
  - Services composable to form new services
    - The role of brokers/integrators

What is new?

Where are the (new) problems?

Closed world

- In a closed world there is a sharp separation between development time and run time

  design/composition
  verification
  deployment

  execution

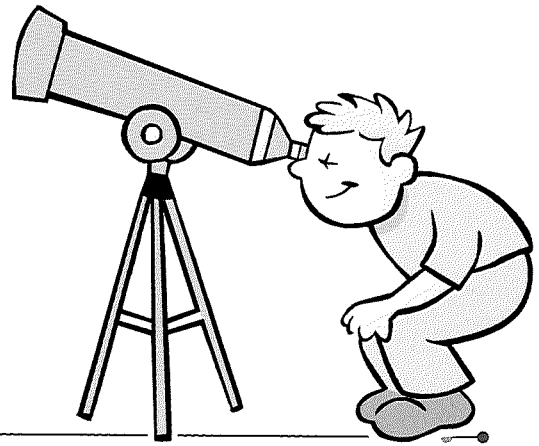
- Changes require switching from run time back to development time where software is changed and validated

Open world

- Dynamic architecture adaptation
  - Both **components** and the **structure** of the composition change
- Continuous validation
  - Dynamic changes and decentralized ownership ask for it

Challenges

- **How can extreme flexibility and dynamic evolution co-exist with high dependability?**
- Architecture and composition mechanisms for open-world software (OWS)
- Validation of OWS
- Specification recovery for OWS



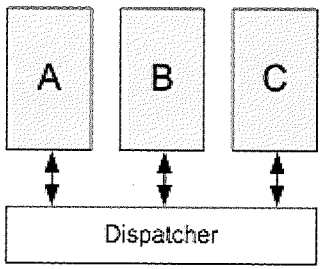
Open world
binding mechanisms

- Implicit
  - via some "logical" coordination space
    - e.g., via events: publish/subscribe
- Two-phase explicit (discovery-based)
  - publication states all functional and nonfunctional properties
  - discovery/selection may involve optimization/ negotiation
- ***By contract?***

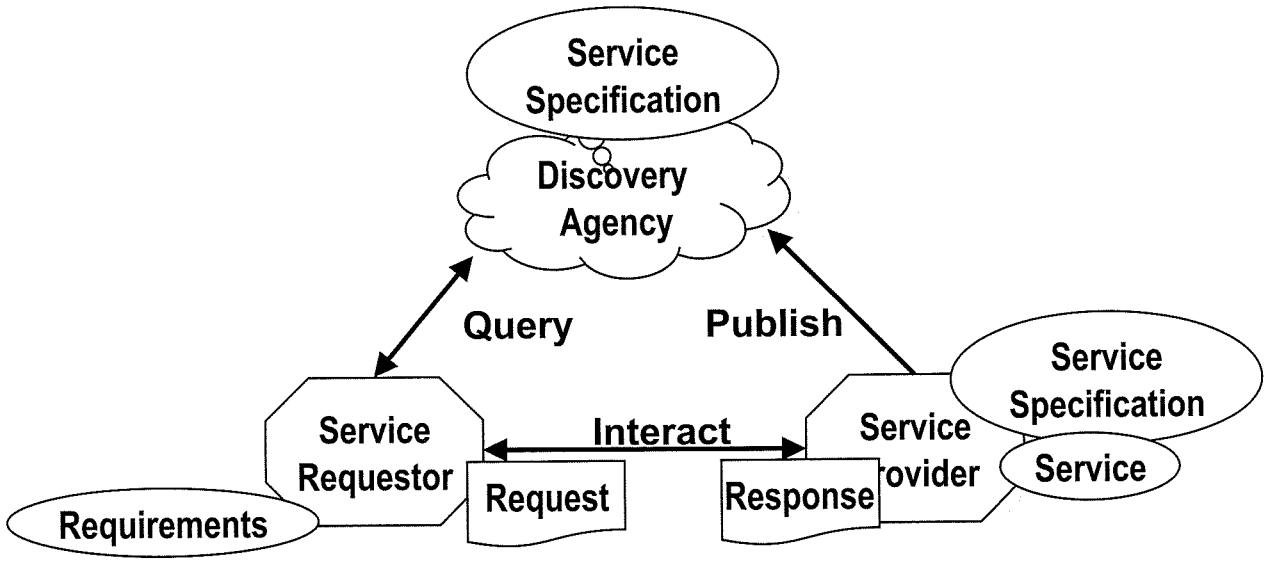
Implicit Binding

the Publish-Subscribe case

- Asynchronous communication mediated by a *dispatcher*
  - anonymous
  - multipoint
  - implicit addressing
    - subject vs. contents-based
- Application components
  - *subscribe* to relevant message patterns
  - *publish* messages
- The dispatcher matches published messages against previously issued subscriptions
- Support dynamic addition, removal and replacement of application components



Discovery-based binding



Binding by contract

interfaces described by a
SPECIFICATION

Specification describes
signature
functional behavior
nonfunctional behavior

We must manage contract violation

Potential benefits

- Clear separation between **what**, **how**, **when**
  - Different policies possible
    - early/late
    - context-aware, situational
    - optimizing service level
      - bind to the "current best"
  - Rich interface descriptions needed
    - contracts must speak about QoS
  - Self-organization

BBC for service compositions

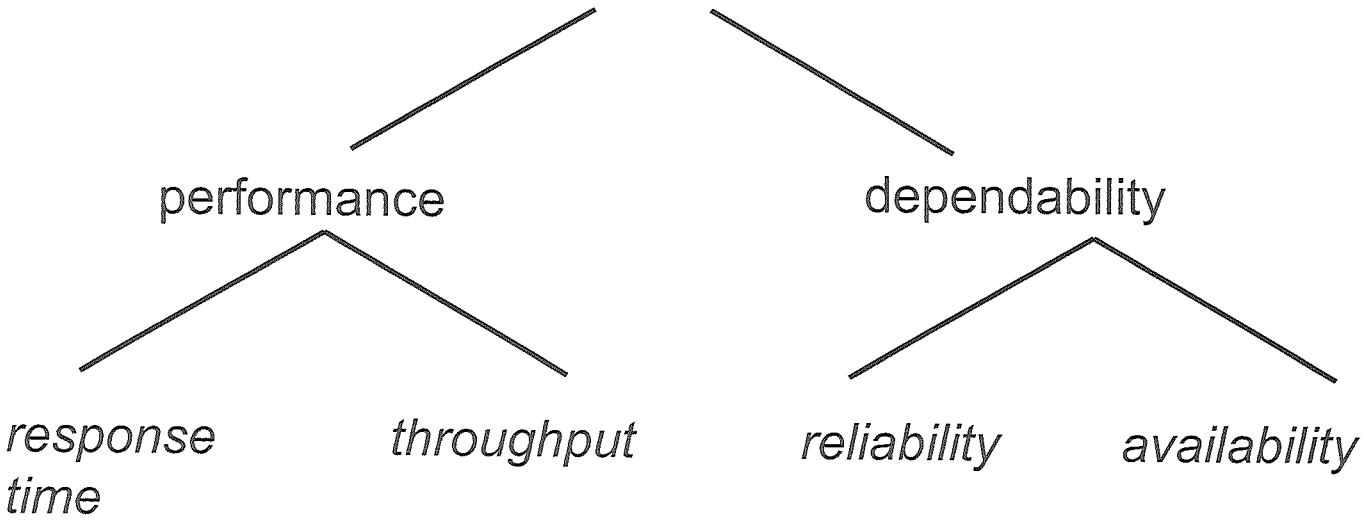
- Workflow-based compositions define an abstract **orchestration**
  - A logically centralized entity acts as an orchestrator
    - BPEL used as a workflow engine
  - External services defined by their rich specs, to support contracts
  - Binding to concrete services may be dynamic
    - To support continuous adaptation to environment and QoS changes

Two kinds of validation

one language

- ALBERT is an assertion language for BPEL processes
  - Can express functional and non-functional properties
- Can be used for two kinds of validation
  - Design-time (model checking)
    - we **assume** that the external world satisfies the stated properties; based on these
    - we **guarantee** certain properties of the composite service
  - Run-time (monitoring)
    - check if properties are maintained
- A single and coherent validation framework

Non-functional specifications

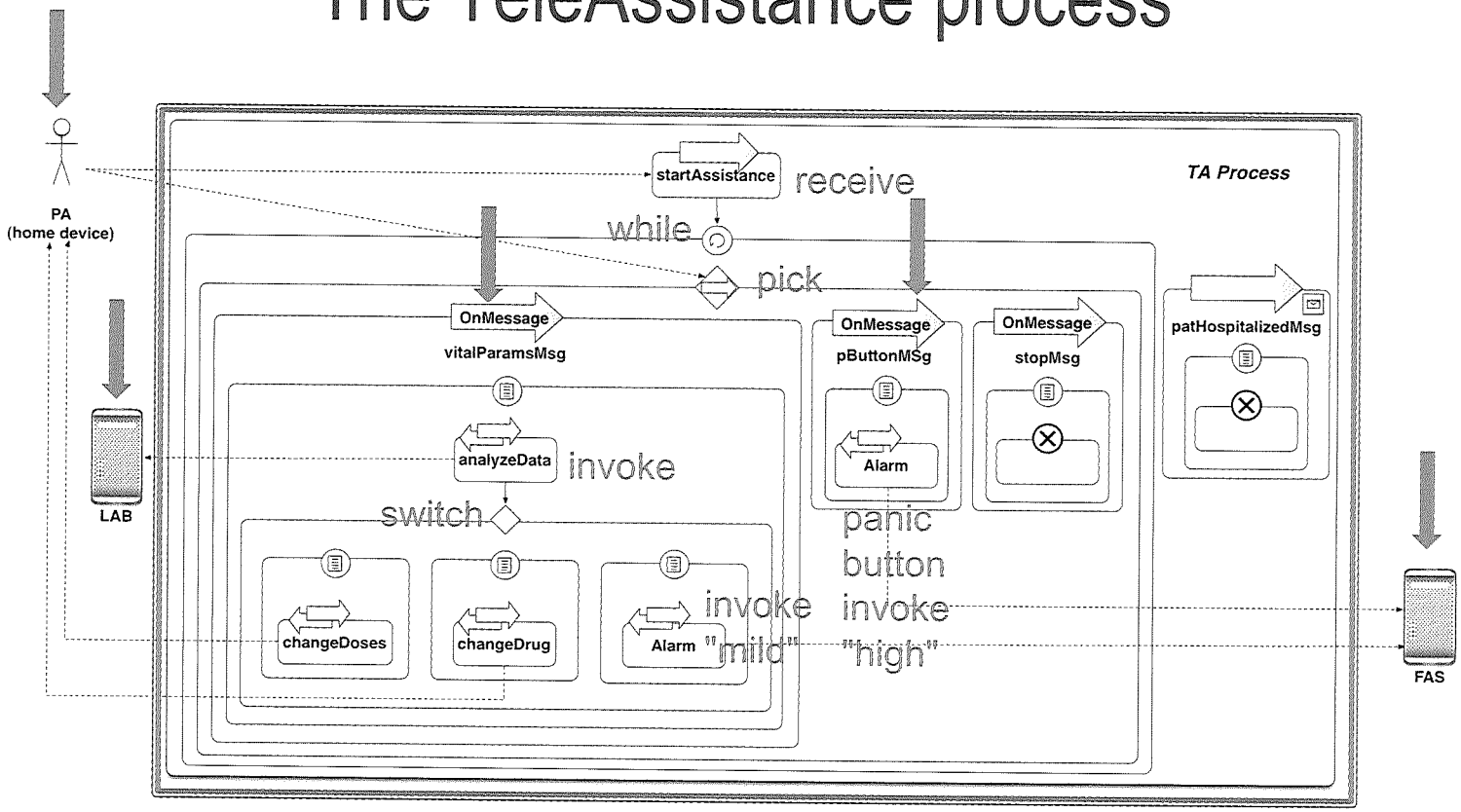


require explicit time and a metric
- distance between time points
- counting (over time intervals)

Assume-guarantee

- A powerful approach to system design and analysis
  - Define what the external world must guarantee
    - **you assume it holds and rely on it**
  - and what you must in turn guarantee
    - **you must prove it holds**
- There is an obligation to eventually **prove/check** that what is assumed about X is indeed guaranteed by X

The TeleAssistance process



Some properties

- **VitalParams**: the glucose value sent by the home device is in range 40..300 mg/dL
- **LabServiceTime**: after sending the patient's data to the lab, it must reply within 1 hour
- **FASConfirmHospitalization**: if the FAS is invoked three times over a week, with a "High" severity level for a certain patient, it must notify the TA, within one day, that the patient has been hospitalized
- **FASInvokeMildAlarm**: after receiving a message from the LAB indicating that an alarm must be issued to the FAS, the TA process must invoke the FAS service within four hours, passing a "Mild" alarm notification
- **MDCheckUp**: if a certain patient sends the pButtonMsg three times during a time span of a week, the FAS must hospitalize the patient within one day

ALBERT

A temporal language that predicates on events and durations
Can express QoS statements
Properties are defined as process invariants

Syntax:

ϕ ::= ψ relop ψ | $\neg\phi$ | $\phi \wedge \phi$ | (op id `in` var ; ϕ) | $onEvent(\mu)$ |
$Becomes(\phi)$ | $Until(\phi,\phi)$ | $Between(\phi,\phi,K)$ | $Within(\phi,K)$

ψ ::= var | ψ arop ψ | const | $past(\psi, onEvent(\mu), n)$ | $count(\phi, K)$ |
$count(\phi, onEvent(\mu), K)$ | fun(ψ, K) | fun$(\psi, onEvent(\mu), K)$ | $elapsed(onEvent(\mu))$

relop ::= $<$ | $\leq$ | $=$ | $\geq$ | $>$

op ::= `forall` | `exists`

arop ::= $+$ | $-$ | $\times$ | $\div$

fun ::= sum | avg | min | max | ...

Properties in ALBERT

- LabServiceTime:

  After sending the patient's data to the lab, it must reply within 1 hour

  onEvent(start_analyzeData) → Within(onEvent(end_analyzeData), 60)

- MDCheckUp:

  If a certain patient sends the pButtonMsg three times during a time span of a week, the FAS must hospitalize the patient within one day

  ∀x(Becomes(count($alarmNotif/level='high' ∧
  x = $alarmNotif/pID, onEvent(pButtonMsg), 10080) = 3)
  → Within(onEvent(patHospitalizedMsg) ∧ $patHospitalized/pId=x, 1440))

Design-time validation

- Given an abstract BPEL workflow and AAs characterizing the external environment, are GAs satisfied?
  - We do validation by model checking, using Bogor
    - The BPEL2BIR environment
  - We analyze all of BPEL (the only tool that does so, as far as we know)

Run-time validation

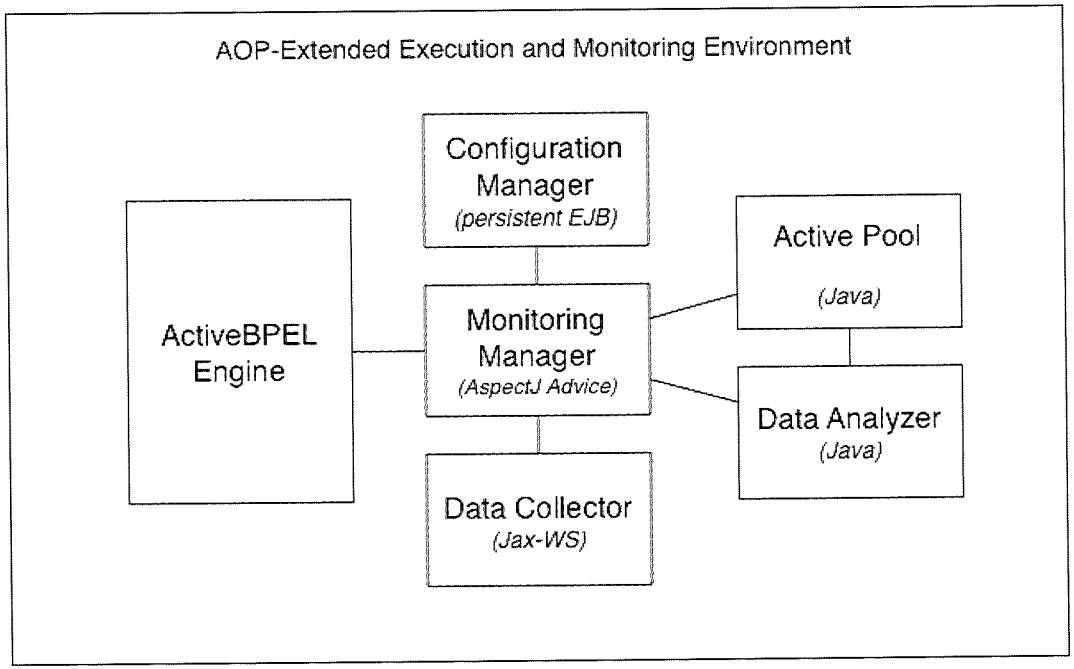
- Two good reasons for doing it
  - Design-time validation is incomplete
    - Model checking
      - exhaustive state exploration, restricted domains
  - Design-time assumptions may be invalidated at run time
    - Due to the decentralized nature of services

•this is specific of open world

Monitoring ALBERT assertions

- AAs and GAs are checked at run time by a monitor
- Data collected and analyzed as the system is running
- Violations may drive suitable reactions, to support a self -managing, autonomic cycle

AOP-extended execution and monitoring



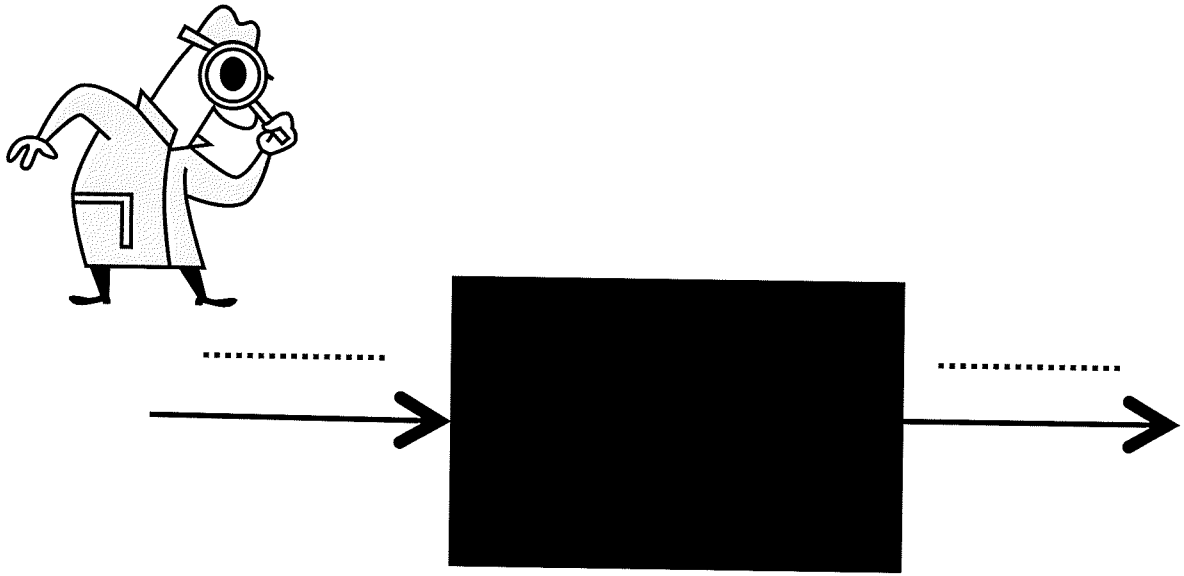
Evaluating ALBERT properties at run time

- Properties that only refer to the present state and/or to the past history
  - history to be kept is bounded
- Properties that refer to the future
  - new evaluation threads are spawned, but a bounded memory is sufficient
- Interaction with ActivePool
  - via its API to retrieve past states
  - by subscribing to new states being generated

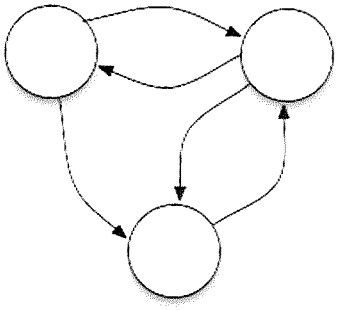
Specification recovery

- Services are searched and selected based on their specification
  - It may be inconsistent wrt to the service
  - It may be incomplete

  Can it be inferred?
- This differs from the traditional reverse engineering problem of extracting high-level views from code
  - Code is inaccessible, only run-time behavior is observable



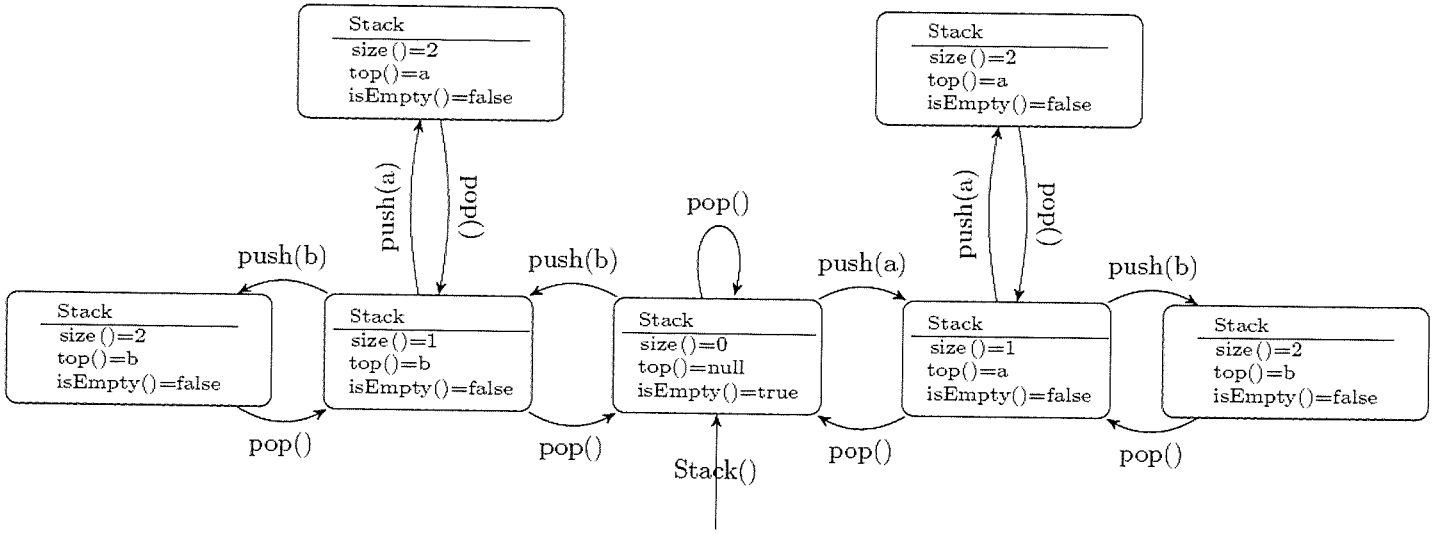
model



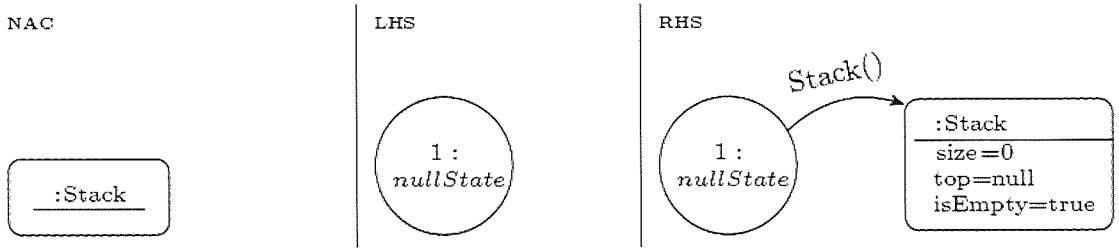
Preliminary results

- Components
  - Reflective interface
  - Data abstractions (Java containers)
- Specifications
  - Algebraic
  - State-machine based
    - Graph transformation rules

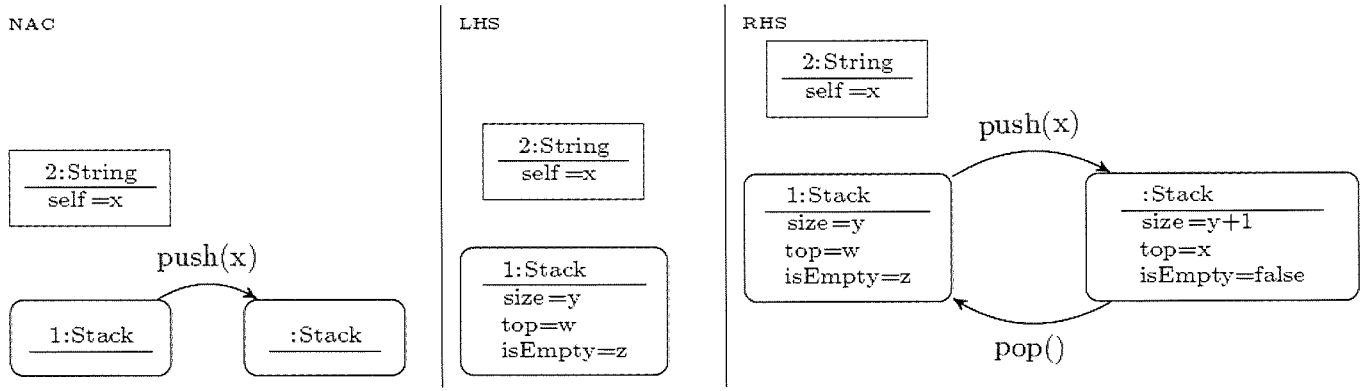
Partial extensional description



Intensional description via GTR



(a) Constructor rule



(b) Push rule

Conclusions

- We are moving to open world software
- Where fundamental problems continue to exist in a new and more difficult form
  - Architecture
  - Verification
  - Specification and specification recovery

Acknowledgements

- On-going projects
  - SeCSE, Plastic, ArtDeco
- New projects
  - S-Cube, Q-ImPrESS
- People
  - L. Baresi, E. Di Nitto, S. Guinea, R. Mirandola, D. Bianculli, P. Spoletini, A. Mocci, G. Tamburrelli

Some papers

- C. Ghezzi, L. Baresi, E Di Nitto "Towards Open-World Software". Computer, IEEE, Volume 39, Number 10: 36-43, October 2006
- L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. Validation of Web service compositions. IEE Proceedings Software, 2007
- L. Baresi, C. Ghezzi, and S. Guinea, "Smart Monitors for Composed Services", 2nd Int.l Conf. on Service Oriented Computing. Nov. 2004
- L. Baresi, C. Ghezzi, and S. Guinea "Towards Self-healing Compositions of Services" In Bernd J. Kraemer and Wolfgang A. Halang (eds.) Contributions to Ubiquitous Computing. Volume 42 of Studies in Computational Intellligence, Springer, 2006
- L. Baresi, E. Di Nitto, C. Ghezzi, S. Guinea "A Flexible Framework for the Deployment of Service-Centric Systems", Service Oriented Computing and Applications, 1, 1, 2007
- L. Baresi, D. Bianculli, C. Ghezzi, P. Spoletini. Monitoring Web Service Orchestrations Using Timed WSCoL Proc. of Int.l Conference on Web Services (ICWS), 2007
- D. Bianculli, C. Ghezzi, Monitoring Conversational Web Services, SOSWE 2007.
- D. Bianculli, R. Jurca, W. Binder, C. Ghezzi, B.Faltings. Automated Dynamic Maintenance of Composite Services based on Service Reputation. In Proc. of Int.l Conf. on Service Oriented Computing (ICSOC), 2007
- L. Zanolin, C. Ghezzi, and L. Baresi. "An Approach to Model and Validate Publish/Subscribe Architectures". In Proc. of the SAVCBS03 Workshop, 2003
- L. Baresi, C. Ghezzi, and L. Mottola. "Towards Fine-Grained Automated Verification of Publish-Subscribe Architectures". In Proc. of the 26th Int. Conf. on Formal Methods for Networked and Distributed Systems (FORTE06), 2006
- L. Baresi, C. Ghezzi, and L. Mottola. "On Accurate Automatic Verification of Publish-Subscribe Architectures". Proc. of the 29th Int. Conf. on Software Engineering (ICSE07), 2007
- L. Baresi, G. Gerosa, C. Ghezzi, and L. Mottola. "Playing with Time in PublishSubscribe using a DomainSpecific Model Checker". In Proc. of the SAVCBS03 Workshop, 2007
- C. Ghezzi, A. Mocci, and M. Monga. "Efficient Recovery of Algebraic Specifications for Stateful Components", In IWPSE '07: Proceedings of the 9th International Workshop on Principles of Software Evolution, September 2007.
- L. Baresi, C. Ghezzi, A. Mocci and M. Monga. "Using Graph Transformation Systems to Specify and Verify Data Abstractions", Proceedings of the 7th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2008).