AMO - Advanced Modeling and Optimization, Volume 16, Number 3, 2014

Evaluating key performance measures of knowledge-based systems via homogeneous stochastic modelling and analysis

Georgios Constantine Pentzaropoulos[1]

Mathematics and Information Technology Unit,
Department of Economics, University of Athens,
1 Sophocleous Street, 105 59 Athens, Greece.

Abstract

Contemporary knowledge-based systems are employed for obtaining intelligent decisions to problem-solving situations by making use of rules and justifications. The performance of KBS's can be characterized by a set of suitable measures. In this work, we consider the following quantities: utilization, availability, and overall system responsiveness. We also create an operational framework for modelling knowledge acquisition from KBS's using arguments from stochastic analysis. Actions to correct instabilities and thus improve KBS performance are also discussed for both centralized and multi-agent systems.

***Keywords*:** Knowledge bases; codification of information; performance evaluation; stochastic analysis; ontologies; measures; information systems.

MSC Classification: 60G07, 68M11, 68P20, 68T30, 94A17.

1. Introduction

Knowledge bases (KBs) are essential elements of many contemporary information economies and societies. They are seamlessly integrated into the store-and-forward nodes of large-scale computer networks acting as digital libraries with sophisticated man-machine interfaces. As such, modern KBs contribute to our understanding of many important subjects of interest while serving as hosts for an increasing number of telematic activities.

KB performance can be characterized by suitable quantitative *measures*: we shall examine in some detail the most important ones in subsequent sections of this article. For the moment, suffice to say that the factors underlying the overall performance of KBs come from two interrelated sources: (i) the KB architecture itself (internal configuration and operational modules) and (ii) the capabilities of the supporting telecommunication infrastructures. The main features of these infrastructures include many forms of broadband technologies (fixed and wireless), access via intelligent man-machine interfaces, and improved quality of service (QoS).

Human-readable knowledge bases, in addition to machine-readable ones, contain implicit knowledge: such kind of knowledge (as opposed to explicit knowledge) is particularly valuable when logical inference does not apply or it is difficult to obtain.

[1]*Associate Professor (ICT). Email: gcpent@econ.uoa.gr.*
AMO - Advanced Modeling and Optimization. ISSN: 1841-4311

The so-called *multi-agent systems* are also part of our examination, since they can integrate the properties of explicit knowledge - also referred to as *tacit knowledge* - with their inherent property of robustness, i.e. their ability to withstand workload pressure much more successfully than the classical centralized systems.

In this article, we proceed as follows. First, we discuss fundamental concepts of knowledge bases making the necessary distinction between machine readable and human readable KBs. Next, we examine the main properties of multi-agent systems in comparison with centralized systems. Then, we go on to more complex subjects as stochastic modelling and analysis in the framework of KB performance evaluation. We conclude with certain practical aspects, which are believed to be essential for the purpose of effective KB management.

2. Machine readable and human readable knowledge bases

Knowledge bases are databases specifically designed for the meeting the needs of knowledge management. They are large information repositories of data which can be searched, utilized, and shared by appropriate user communities. There are two kinds of knowledge bases: machine-readable and human-readable KBs.

Machine-readable knowledge bases (MRKBs) contain data and rules presented in a way by which they can be logically read by another machine; hence, no human intervention is necessary. Knowledge contained in this type of bases is explicit by design and tractable by means of logical inference. Artificial intelligence (AI) techniques and algorithms are employed here [1]. Knowledge representation and reasoning are central themes in this design which also involves the encoding of given propositions. AI practice is concerned with the construction of knowledge-based systems realized via suitable man-system interfaces. Logical operators such as (AND, OR) are used in the course of machine interpretation. Such knowledge bases are extensively used within the so-called semantic web [2,3,4].

Human-readable knowledge bases (HKRBs) differ from the above type as they also contain implicit knowledge: this kind of knowledge - known as tacit (more than can be said) - is particularly valuable when the formal rules of logical inference are not applicable in a given situation. User content may refer, for instance, to banking or medical history or to learning activities. In the business world, content may also include articles and reports, user manuals, and other documents for sharing amongst workers and their clients. Search engines are the means for retrieving information of interest and for relaying that information, thus facilitating *knowledge exchange*. Intranets are also frequently used in such circumstances.

Figure 1 shows the main elements of a knowledge base in the world-wide web along its user population. Note the presence of an inference engine associated with the KB in question as well as the interface connecting the users. The query-reply format is typical in the case of a human-readable knowledge base. When necessary, the contents of a machine-readable knowledge base may be converted into a natural language format so that humans can understand the facts and associated rules embedded there. Semantic web languages are often used for the above purpose.

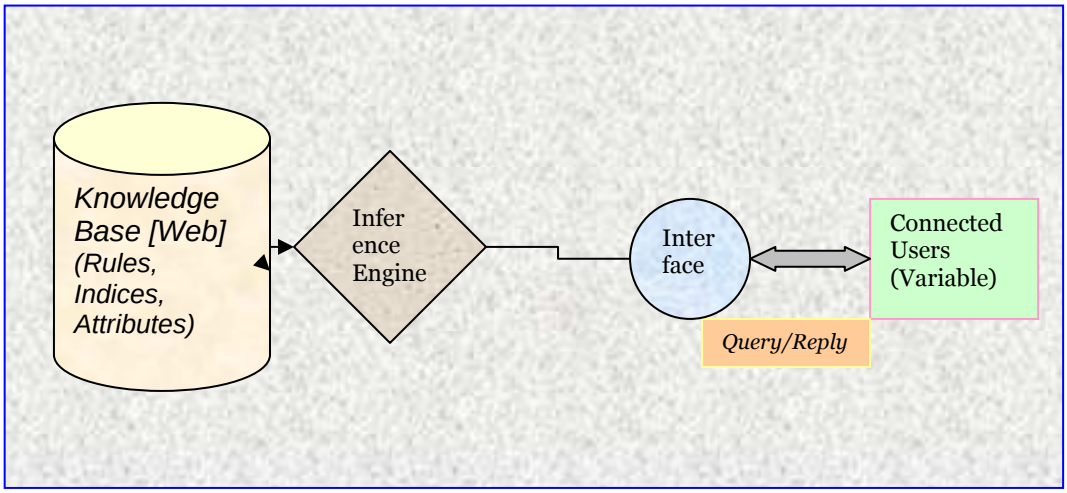

Figure 1: Elements of KB-user interaction in an open communications environment.

When the specification language used is close to natural language, the transformation from a machine readable KB to a human-readable one is made easier. The TIL-script programming language, for example [5], is capable of encoding the content of ontologies and other attributes. TIL-script messages are presented in a standard natural language that is very close to needs of human agents.

Knowledge-based systems (KBS) are "products" of artificial intelligence (AI). They are employed for the purpose of providing intelligent decisions to problem-solving situations by making use of rules and justifications. At its core, a KBS contains a large amount of information as well as an elaborate set of concepts, assumptions and rules. Further, a reasoning system implemented in the KBS helps in making intelligent decisions. Any knowledge base, whatever its type, makes use of the concept of ontologies. An ontology is simply a set of attributes assigned to the objects of a KBS, and to the inter-relations of these objects. Thus, KBS are able to support human learning, enhance understanding, and provide guidance for action.

The inference engine of Figure 1, upon the receipt of a request, searches the knowledge base and then applies all relevant rules and relationships amongst objects. It also processes the associated information encoded there and provides the necessary justification for further action. Apart from its internal structure, KBS performance also depends on the workload received during a period of observation. Since such workload is always a function of the user population - more precisely the subset of active users - connected to the knowledge base, it follows that the overall KBS performance is also influenced by workload fluctuations. We examine this matter later in this article.

For the time being we simply note that, when user activity increases, system (KBS) responsiveness $R(s)$ decreases proportionately [6]. And in the limiting case, i.e. when $N \rightarrow \infty$, which practically means that N has exceeded some critical point, it follows that $R(s) \rightarrow 0$. This is seen by the users as inability to communicate with the distant KBS host system [7]. Also, when system *entropy* increases, instabilities occur leading to an increase in uncertainty which, in turn, affect the sequence of logical inferencing and the quality of decisions [8].

3. Stochastic modelling and analysis

Analysis of complex systems can be performed by means of either deterministic or probabilistic models. Because of their known potential for representing random phenomena in a compact manner, probabilistic models are preferable in the analysis of computer systems and networks. The same potential is present when one attempts to study the performance of databases or, in the present situation, knowledge bases which are distributed across web connections. Amongst the probabilistic approaches to the modelling of of the above type of knowledge bases, the queueing approach is by far the most appealing one because of its versatility and robustness.

This section begins with a brief account of the structure and properties relating to stochastic processes and Markov chains. Then, we introduce a modelling scheme of knowledge acquisition from a web knowledge base in the framework of this study. The key performance parameters of this model are next derived analytically followed by an example illustrating our approach. Finally, we discuss the results obtained in the context KBS performance management.

3.1. Stochastic processes and Markov chains

A stochastic process $X(t)$ is a function of time t whose values are random variables. Further, a Markov chain is a stochastic process $X(t)$ with states $S_0, S_1, \ldots S_i \ldots$, such that the probability at time t_{k+1} an arbitrary state S_i depends only on the state at time t_k for any sequence of time instants $t_1, t_2, \cdots t_{K+1}$ with $t_1 < t_2 < \cdots t_{K+1}$. The probability of a transition from state S_i to state S_j at time k may be written as follows:

$$p_{ij}(k) = \text{prob}\{X_{K+1} = j \mid X_K = i\}. \qquad (1)$$

The statistical relationships amongst the possible states of a Markov chain can be specified by means of a matrix $\mathbf{P}(k)$ known as the transition-probability matrix. Further, we make the assumption that the transition probabilities can for practical purposes be independent of time; therefore, our chain is assumed to be *homogeneous.* The transition matrix $\mathbf{P}(k)$ of a homogeneous chain can be graphically illustrated when the state space is moderate.

Stochastic models including homogeneous chains can be used with confidence for the modelling of real systems. They are especially useful in the study of transitions amongst possible states. With such models the analyst is able to answer several performance-related questions. Examples of such questions are as follows:

(i) How often is a certain state visited in the course of an observation period?
(ii) How much time is spent in that state by system under examination?
(iii) How long - or short - are the intervals between successive visits?
(iv) Are the operating overheads sufficiently low or at least tolerable?

If our model satisfies certain conditions which are detailed below, then answering questions such as the above is possible. Of course, one needs to have closed-form analytical expressions in order to avoid lengthy simulation procedures.

3.2. Equilibrium state probabilities

Analytical expressions require the adoption of the following properties which often characterize homogeneous chains:

P1: *Irreducible chain.* This property is considered true when a state can be reached from any other state.
P2: *Recurrent states.* A state is considered recurrent if the probability of re-visiting it, after a visit has already taken place, is equal to 1.
P3: *Aperiodic states.* If the visits to a recurrent state have recurrence times that are not all equal, then that state is aperiodic; and if all states are aperiodic, then the entire Markov chain is aperiodic also.

Let us denote by $p_i\,(k)$ the probability that a Markov chain, at time k, is in some state S_i. The initial state probabilities are simply $p_i\,(0)$. Let us also recall our initial assumption about homogeneity and then combine this with properties P1, P2 and P3. Then, the limiting state probabilities:

$$p_i = \lim \{p_i\,(k)\} \text{ with } (k \rightarrow \infty,\ i = 0, 1, 2, ...) \qquad (2)$$

exist and are independent of the initial state probabilities.

Assuming that all mean recurrence times are finite - which is thought realistic in most system observations - then the state probabilities are merely a stationary series, and they can be found analytically be solving the following set of equations:

$$p_j = \sum_i p_i\, p_{ij} \quad \text{with } (j = 0, 1, 2, ...) \qquad (3)$$

and

$$\sum_i p_i = 1. \qquad (4)$$

The solution of Equations (3) and (4) above give the equilibrium state probabilities which are independent of the initial state probabilities. From this set of equations we can compute several key performance measures which are of interest in this study. Before going into the details of such computations we need to discuss the structure of our model and relate it to the performance study of remote knowledge-based systems. So, let us consider the following figure:

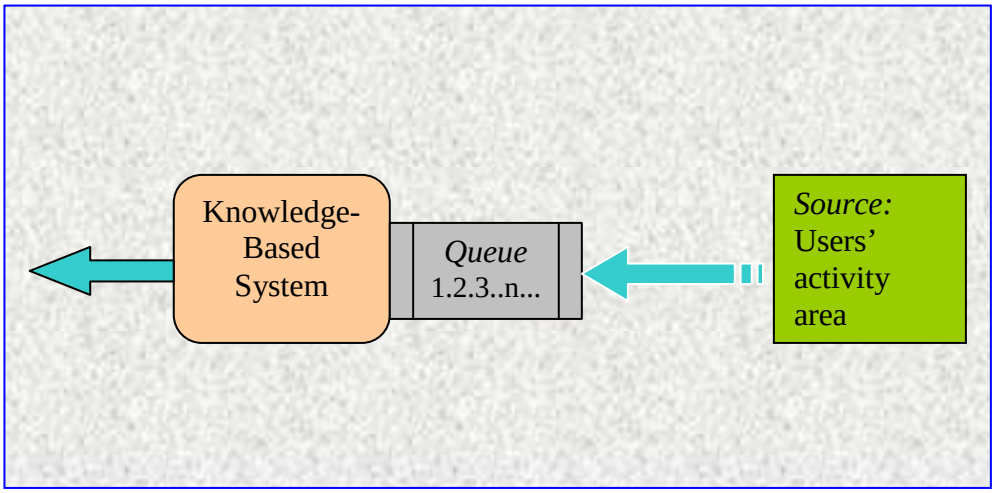

Figure 2: Model of users' requests for service. Requests are placed in a queue.

The figure comprises the following elements: (1) a KBS with any of the two types of knowledge bases (MR/HR) mentioned in the previous sections; (2) a physical source where system users make their requests; and (3) a queue in which the above requests are placed awaiting service. When service is completed, an output for each request is generated and this output follows the direction of the arrow on the left.

Our interest is focused on certain key performance measures that can be used in the analysis of the KBS: further, we seek closed-form expressions for these measures so as to avoid simulation or lengthy iterative procedures. Such performance measures characterize KBS operation in terms of (i) resource utilization, (ii) responsiveness, and (iii) stability. First, we need to specify all possible state. From Figure 2 above, and by taking account of all KBS elements, we propose the following state space:

- S_0: The *waiting state,* i.e. when the KBS expects input from the its users.
- S_1: This may be called the *user state*, which corresponds to all active users.
- S_2: The *scheduling state,* which is associated with queueing and internal functions.
- S_3: Finally, the *problem-solving state,* i.e. the overall service provided by the KBS.

From the above state space we can construct the transition-probability matrix **P***(k)* by giving some values as regards possible state transitions ($S_i \leftrightarrow S_j$). Such an example with indicative values follows below:

Table 1: Transition-probability matrix for the example KBS

States	S_0	S_1	S_2	S_3
S_0	0.97	0.03	0.00	0.00
S_1	0.03	0.93	0.04	0.03
S_2	0.00	0.03	0.91	0.04
S_3	0.00	0.02	0.04	0.96

First, we note that the equilibrium state probabilities must always sum up to value 1 according to the requirement of Equation (4) above. Next, by solving Equations (3) for the values of Table 1 we obtain:

$$p_0 = 0.23; \quad p_1 = 0.12; \quad p_2 = 0.07; \quad p_3 = 0.58. \tag{5}$$

3.3. Utilization and time intervals

From the above results we may obtain the degree of *utilization*, i.e. the fraction of busy time of our KBS as the complement of p_0 above. Calling this fraction ρ we see that $\rho = (1 - 0.23) = 0.77$, or in percentage form $\rho = 77$ %. This implies that $\rho < 1$, which is expected because our system operates within its equilibrium. Therefore, the inequality $\rho < 1$ can be seen as a necessary condition for system *steady-state.*

Utilization is an important measure as it shows how busy is an operating system during a period of observation. When $\rho \rightarrow 1$, this is an indication that our system, here the KBS, has the tendency to leave its equilibrium, thus drifting into instabilities and possible saturation. We examine these matters in the next section.

For the time being, we can also calculate another useful measure which may be called the *mean duration* of state S_j (j = 0, 1, 2, 3, ...). At every time instant, starting with state S_j, our chain has a probability p_{ij} of remaining there and a complementary probability $(1 - p_{ij})$ of going to another state. The properties of the previous seb-section (P1, P2, P3, P4) and the homogeneous nature of our chain allow us to calculate the mean duration times as follows.

Let us first assume that, when a state S_j (j = 0, 1, 2, 3, ...) is active, it may last for a maximum of q milliseconds. The quantity q may be called *quantum* or time-slice. Such very short amounts of time are frequent in real situations, especially whenever service is offered on on a time-sharing basis. Our KBS is a good example of a system operating on the above basis since it has to cater for the needs of many users with different demands (see also Figure 2). Then, the mean duration T_j of state S_j can be calculated as follows:

$$T_j = q \,/\, \{ (1 - p_{ij}) \} \; (j = 0, 1, 2, 3, ...). \tag{6}$$

Assuming a quantum q = 75 ms, application of Equation (5) above, in connection with the values of p_{ij} in Table 1, gives the following values (in seconds): $T_0 = 2.500$; $T_1 = 1.071$; $T_2 = 0.833$; $T_3 = 1.875$.

We note that, on the average, most of the time is spent in states S_1, S_2 and S_3 which characterize user, scheduling, as well as problem-solving states. An adequate amount of time is also spent in state S_0, i.e the user waiting state. Therefore, we may note that our example KBS system is stable, capable of serving its users efficiently.

4. System performance and instabilities

Performance instabilities principally occur when user activity gradually increases and becomes intense thus approaching the KBS ability for service. This phenomenon can be illustrated by recalling the definition of state S_0 of the previous section. S_0 is the *waiting state*, i.e. the state in which the KBS expects input from its users. Associated with S_0 is p_0, i.e. the corresponding equilibrium state probability.

4.1. User activity and system utilization

The impact of such intense user activity on system utilization can be illustrated by allowing p_0 to gradually decrease, as shown in Table 2. The sum of the rest of the probabilities also increases so as to keep to the sum total equal to 1 according to Equation (4) of the previous section. System utilization ρ is the complement of p_0 as above, thus ρ will increase proportionately. The effect of these changes is shown in Table 2. When ρ reaches the value of 95%, we may say that our system is approaching its *saturation point.*

Table 2: Progress of utilization for the example KBS

p_0	$p_1+p_2+p_3$	ρ (%)
0.23	0.77	77
0.20	0.80	80
0.17	0.83	83
0.14	0.86	86
0.11	0.89	89
0.08	0.92	92
0.05	0.95	95 *(*)*
0.02	0.98	98
0.00	1.00	100 *(sat.)*

(*) Onset of system saturation.

This simple numerical example illustrates the general requirement for *steady-state* briefly referred to earlier, i.e.:

$$\rho < 1 \text{ (steady-state) and } \rho \rightarrow 1 \text{ (saturation)} \qquad (7)$$

When $\rho = 1$, our KBS becomes completely saturated, thus unavailable to its users. Another way of looking into this troublesome situation is to observe that when $\rho = 1$, then $p_0 = 0$. Recalling from earlier the definition of the waiting state S_0, we see that in this case the system is working at its full capacity. In classical queueing theory ρ is also known as the system *traffic intensity.* Then, ρ relates arrival rates to the system service rates. For stability, it is required that values of the arrival rates are *always less* than the corresponding values of service rates.

4.2. System availability and responsiveness

Apart from system utilization, another important measure that characterizes a KBS is its *end-to-end delay*. This measure may be defined as the sum of the times the system works for its users, i.e. all intervals of time except the interval corresponding to the waiting state S_0. The above measure, which we denote by T_{sys}, takes into account all possible delays attributed to the queueing of users' requests for service as well as the delays associated with network transmissions.

The value of T_{sys} above can be calculated by virtue of the well-known Little's law from classical queueing theory. Little's law, originally formulated by J.D.C. Little as a theorem, has been proved in the course of time to be true for any system with arbitrary arrival and service times. Stated informally, Little's law equates the number of "customers" N to the product of the system's arrival rate λ and the corresponding waiting time W. Hence, the well-known equation: *"$N(s) = \lambda \cdot W$"* where the above quantities are expressed by their mean values. W includes both the "customers" being served and those waiting in the queue.

Figure 2 given earlier is an appropriate model in this situation provided that we first translate "customers" to users' requests for service, then equating W to T_{sys}, and finally relating the arrival rate λ with system workload and responsiveness. The latter relationship is given analytically from classical queueing theory as the ratio of λ to the system's service rate μ. Hence, the traffic intensity formula $\rho = \lambda/\mu$.

From this formula and by expressing the service time $1/\mu$ as the value of the quantum (q), we can see that $\rho = \lambda \cdot q$. Finally, by remembering that ρ and p_0 are complimentary quantities, $(1 - p_0) = \lambda \cdot q$. From this expression we can calculate the value of λ as follows:

$$\lambda = (1 - p_0) / q. \tag{8}$$

Application of Little's law with the above notation gives the following relationships:

$$N(s) = \lambda / (\mu - \lambda) ; \quad T_{sys} = 1 / (\mu - \lambda). \tag{9}$$

Let it be noted that Equations (8) above require the additional assumption of Poisson arrivals λ and exponentially distributed service times $1/\mu$. This type of queueing system is denoted by M/M/1/∞ in Kendall's notation. Finally, the probability:

$$P \text{ (no access)} = \text{prob } \{N \geq k\} = \rho^k \tag{10}$$

may be used to show the case when the number of requests N exceeds some critical value k of the system's capacity. Applying our example values, we get: λ = 0.77/75 msec = 0.77/0.075 sec = 10.27 requests/sec, from which $\rho = \lambda/\mu = \lambda \cdot q = 10.27*0.075 = 0.77$ or ρ =77%. Since, $\rho < 1$, our example system is in its steady-state.

From Equation (10) we can estimate the *availability A(s)* of our system, i.e. the fraction of users' requests that can be accepted for service given the system's internal capacity limitations. Then, for a value of e.g. $k = 10$, i.e. a maximum of 10 requests allowed in the system (waiting for or receiving service), we find that: *P (no access)* = prob $\{N \geq 10\} = 0.77^{10} = 0.073$. Therefore, 7.3% of the incoming requests will have to be denied access. Thus, system availability stands at *A(s)* = 92.7%.

Equations (9) and (10) above are very useful in exploring the consequences of gradually increasing workloads. This happens when and $\rho \rightarrow 1$ as in expression (7) or equivalently when $\lambda \rightarrow \mu$. Table 3 below shows the performance of our example system as the function of its workload for a value of $k = 10$.

Table 3: Availability and responsiveness for the example KBS

λ (req./sec)	*A(s)* (%)	*N(s)* (num)	T_{sys} (seconds)
10.27	92.7	3.39	0.33
10.67	87.9	4.06	0.38
11.07	84.5	4.87	0.44
11.47	77.9	6.19	0.54
11.87	68.8	8.07	0.68
12.27	56.6	11.54	0.94
12.67	40.1 *(*)*	19.26 *(*)*	1.52 *(*)*
13.07	18.3	50.32	3.85
13.33	0.0 *(sat.)*	*(∞)*	*(∞)*

4.3. Instabilities and saturation

From Table 3 we can clearly see that, after the *(*)* point in *A(s)*, system availability drops below 50% and thereafter it declines very sharply. This *(*)* point corresponds to its equivalent onset saturation point, in Table 2 shown earlier, where $\rho = 95\%$. The last line of Table 3 shows a completely saturated system, which again from Table 2 corresponds to $\rho = 1$. This is the effect of an increasing workload which gradually approaches and then reaches the capacity of the example system. The last two columns of Table 3 show this effect in connection with the number of requests being served and the corresponding times within the system.

Again we can clearly see the effect of workload increase by noting the values of the last three rows of Table 3 above. After the *(*)* point, the number of requests being served increases exponentially and the same is true with the times required to complete service. For example, the value 1.52 *(*)* seconds is $(1.52/q) = (1.52/0.075) = 20.27$, i.e twenty times over the system quantum: a very sharp increase. Also, at the same time, *N(s)* is nearly twenty requests within the system, and this number also grows exponentially after the *(*)* point. Finally, the last row of Table 3 above shows the system at its complete saturation state, hence the infinite values of *N(s)* and T_{sys}.

Let it be noted that in critical situations, e.g. when the workload brought forward to a KBS approaches its inherent capacity, overall performance degrades, the system becomes unstable and it may eventually collapse. This unpleasant phenomenon is

known as *thrashing,* a term originally given by Denning and Buzen in the context of centralized systems [9]. The performance of such systems has also been investigated by many authors in the setting of knowledge-based systems which may be seen as dynamical systems operating across the Internet for the purposes of learning and discovering hidden information [10, 11, 12].

The newer multi-agent systems are generally considered as more efficient and flexible than classical centralized systems. They are constructed as autonomous, intelligent agents with considerable communication capabilities. When a multi-agent system is in operation under the same critical conditions, some of its agents can withstand the pressure and operate almost normally. Figure 3 below shows the performance curves of a centralized system (CS) and that of a multi-agent system (MAS) as functions of the system workload over time.

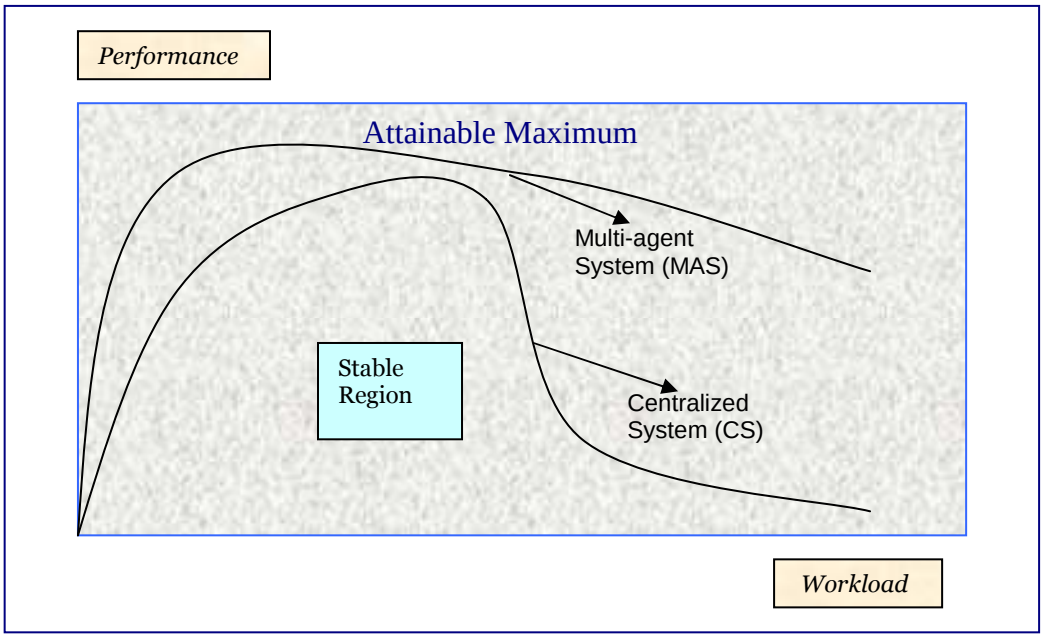

Figure 3: Performance curve as a function of workload for two systems (CS, MAS).

Both systems in Figure 3 operate quite normally within their stable regions attaining their maximum performance as they approach their *plateau,* i.e. the top portion of their respective curves. Thereafter, they both enter their unstable regions. In the previous analysis, we have shown analytically how the increasing workload affects system responsiveness *A(s),* also noting the sharp increase in the values of both *N(s)* and T_{sys}. The main advantage of implementing a multi-agent system is that overall performance - seen as a synthesis of the above three measures - degrades at a much slower pace, thus allowing analysts and installation managers to take corrective actions in a timely fashion. Such actions may include the implementation of control mechanisms in the users' area to limit arrival rates when necessary (*ex ante* actions) or upgrade parts of system configuration so as to make the whole system more responsive to users' requests (*ex post* actions).

Problems concerning performance evaluation, control, and system organization with options for capacity planning are discussed in detail in [13, 14, 15] and in their related bibliography, in the broader context of computer/communication systems. Issues of performance analysis and optimization for systems supporting many classes of *"customers"*, i.e. tasks requiring service from distant hosts, are analyzed in [16]. Finally, backlog-controlled systems with possibilities for their performance control and improvement are studied in [17].

5. Concluding remarks

The problem of performance evaluation and modelling of knowledge-based systems has been approached, in the present work, from a practical/operational point of view. We have developed a framework for modelling knowledge acquisition from KBS's using arguments from stochastic analysis. Especially helpful were queueing-theoretic results such as Little's Law in conjunction with analysis of transition probabilities concerning possible system states.

Key performance measures derived analytically for a given example KBS were: (1) utilization ρ, (2) availability $A(s)$, and (3) overall system responsiveness T_{sys}. Performance instabilities, when traffic intensity increased, were highlighted both graphically and analytically. When the workload brought forward to the example KBS approached its inherent capacity, overall performance was clearly seen to degrade sharply. Thereafter, the system became unstable, eventually reaching its saturation point. For the example values used, it has been found that - at the onset of system saturation - T_{sys} was nearly twenty times over the system quantum (q), which is a very high increase.

Also, at the same time, the number of requests $N(s)$ being served by the KBS grew exponentially. Thereafter, the KBS entered its unstable region (as shown in Figure 3) eventually becoming saturated, thus unable to serve its users. The infinite values of $N(s)$ and T_{sys} (in Table 3) show this progression.

Actions to correct instabilities and thus improve system performance were also discussed with reference to centralized and to the newer class of multi-agent systems. The implementation of control mechanisms in the users' area to limit arrival rates when necessary or the upgrading of system configuration can both make the whole system more responsive to users' requests. Finally, it has been noted that modern knowledge-based systems, accessed via the Internet, are essentially a special class of dynamical systems and, as such, are always susceptible to performance instabilities.

Acknowledgements

This work has been partially financed by grant "elke/70/11/698" awarded by the Research Committee of the University of Athens, Greece. Also many thanks are due to Dr. Neculai Andrei, Chief Editor of AMO, for his excellent cooperation.

References

[1] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach, 3rd Edition.* Prentice-Hall, New Jersey.

[2] Flake, G.W., Pennock, D.M. & Fain, D.C. (2003). The Self-Organized Web: The Yin to the Semantic Web's Yang, *IEEE Intelligent Systems,* 21 (7), pp. 72-86.

[3] Berners-Lee, T. (2005). *The Fractal Nature of the Web: Working Draft.* Available at: (www.w3.org/DesignIssues/Fractal.html)

[4] Shadbolt, N., Hall, W. & Berners-Lee, T. (2006). The Semantic Web revisited. *IEEE Intelligent Systems,* 21 (3), pp. 96-101.

[5] Dusi, M. & Materna, P. (2009). Concepts and Ontologies. In: Kiyoki, Y. et al. (Eds.) *Information Modelling and Knowledge Bases. IOS Press,* The Netherlands, pp. 45-64.

[6] Ferrari, D. (1978). *Computer Systems Performance Evaluation.* Prentice-Hall, New Jersey.

[7] Pentzaropoulos, G.C. (2013). Limits of responsiveness for geographically remote knowledge bases: an operational analysis. *Advanced Modelling and Optimization,* 15 (2), pp. 249-260.

[8] Pentzaropoulos, G.C. (2014). Conceptual framework for modelling knowledge acquisition via content-related Internet sources. *Advanced Modelling and Optimization,* 16 (1), pp. 199-210.

[9] Denning, P.J. & Buzen, J.P. (1978). Operational analysis of queueing networks. *ACM Computing Surveys* 10 (3), pp. 225-261.

[10] Yoshizumi, H., Hori, K. and Aihara, K. (2000). The dynamic construction of knowledge-based systems, in: C.T. Leondes, (Ed.), *Knowledge-Based Systems: Techniques and Applications,* Academic Press, California, pp. 560-604.

[11] Fan, L. (2010). Web-based learning support systems, in: J.T. Yao (Ed.), *Web-Based Support Systems,* Springer, London, pp. 81-94.

[12] Sweeney, E., Curran, K. & Xie, E. (2010). Automating information discovery in the invisible web, in: J.T. Yao, (Ed.), *Web-Based Support Systems,* Springer, London, pp. 167-180.

[13] Lazowska, E.D., Zahorjan, J., Graham, G.S. & Sevcik, K.C. (1984). *Quantitative System Performance,* Prentice-Hall, New Jersey.

[14] Menascé, D., Almeida, V. & Dowdy, L. (1994). *Capacity Planning and Performance Modelling,* Prentice-Hall, New Jersey.

[15] Cassandras, C.G. & Lafortune, S. (2008). *Introduction to Discrete Event Systems, Second Edition,* Springer, New York, USA.

[16] Chen, R-R & Meyn, S. (1999). Value iteration and optimization of multiclass queueing networks, *Queueing Systems,* 32 (1-3), pp. 65-97.

[17] D.I. Giokas and G.C. Pentzaropoulos (2000). Efficient storage allocation for processing in backlog-controlled queueing networks using multicriteria techniques, *European Journal of Operational Rerearch,* 124, pp. 539-549.