Discrete Event Dynamic Systems: Theory and Applications, 12, 253–264, 2002

Unity in Diversity, Diversity in Unity: Retrospective and Prospective Views on Control of Discrete Event Systems

Report on WODES2000 panel discussion, prepared by R. Boel.
Panel members:

X.-R. CAO
Hong Kong University of Science and Technology

G. COHEN
ENPC, Paris

A. GIUA
Università di Cagliari

W. M. WONHAM
University of Toronto

J. H. VAN SCHUPPEN
Chairman, CWI, Amsterdam

Introduction

At the turn of the millennium, in August 2000, the 5-th Workshop on Discrete Event Systems, WODES2000, was held in Ghent, Belgium. This was a very appropriate occasion to bring together some of the eminent researchers who were present there for a panel discussion about the past and the future of the field to which they contributed so much. Four top researchers, who have taken part in the development of the field of discrete event systems and control from its early start, were willing to reminisce about its origins, about its past successes, about the shortcomings of the past work, and about the research that is needed in order to remedy these shortcomings in the (near?) future.

The panel members were Xi-Ren Cao (Hong Kong University of Science and Technology), Guy Cohen (ENPC, Paris), Alessandro Giua (Università di Cagliari), and W. Murray Wonham (University of Toronto), while Jan H. van Schuppen (CWI, Amsterdam) accepted to chair this discussion session. Geert Stremersch made an audio tape of this discussion. This report on that panel discussion is based on listening several times to this audio tape, on my own recollections and on notes taken during the discussion, but also and very importantly on written notes provided by the four panel members. For the sake of uniformity I have taken the liberty to rephrase their text. I hope that the four panel members will recognize their opinions in the following text, and will not disagree too much with the ideas expressed below. Of course I am fully responsible for any inaccuracies, or for any misunderstandings that might result from this text.

The discussion was kicked off with position statements by the four panel members who looked at the questions about past and future, about achievements and challenges of discrete event systems, each from their own different perspectives. The audience, a large part of the WODES2000 registrants, afterwards took part in this discussion in a frank and vigorous way. They played a major role and provided important contributions to the discussion.

In writing down this report I took the liberty to re-order the flow of the discussion in order to bring together themes broached by different participants at different times. This report starts with the retrospective thoughts of the panel members on the origins of the field, followed by a section on the required future developments as seen by the panel members. Many of the questions and comments that were raised during the discussion afterwards were directly related to these statements, and I have summarized most of this discussion, as far as it was related to theory and applications, in this part of the text. An important issue during the discussion was the fact that the inclusion of DES control in engineering education as well as the development of easily used tools are both prerequisites for further progress of applications as well as of theory.

The closing comments by the panel members were surprisingly unanimous: the field of discrete event dynamical systems, with its many different paradigms, is too young to settle on one single universal paradigm. But common problems and common goals, as well as common methodology, compel us to learn from each other and to understand each other's language, justifying the title of this report: unity in diversity, diversity in unity.

The Past Successes?

When the modern approach to system theory was developed by R. E. Kalman and others in the 1960s, it was clearly accepted that the state space approach was related to automata theory. The book of Kalman et al. (1969) provides us with a good view of the way this connection between automata and dynamical systems was understood then. The success story of control in the 1960s and 1970s though was in putting man on the moon, using continuous, even ''linear'' methods. It took more than 20 years before the digital revolution, the computer-based control technologies, the IT explosion, and the pervasiveness of large, man-made systems, forced control theorists to once again explore the relation between finite-state automata and dynamical systems and control. And it has taken even more time for these developments to filter through to applications. The members of the panel started the discussions by reminiscing about how they came to feel the need for introducing a discrete event dynamical systems (DES) concept, starting from their systems theory background.

The seminal work of Ramadge and Wonham (1987) in the early 1980s is one of the approaches that led many of us control engineers towards DES control (even though the term discrete event dynamical systems—DEDS—was probably coined by Larry Ho, who unfortunately could not be present for this panel discussion). During the retrospective part of the panel discussion Murray Wonham emphasized that the supervisory control paradigm he introduced with Peter Ramadge aimed at providing a comprehensive and structural treatment of the control of dynamical systems represented by automata. Supervisory control theory aimed at a qualitative treatment with a control flavor of the discrete world.

As in classical system theory Ramadge and Wonham (1987) used the duality between a state-based formulation and a linguistic formulation. The supervisory control paradigm uses concurrent discrete event models of the components of the possibly very large plant to be controlled—the open loop plant model—and makes explicit the specifications to be satisfied by the closed loop consisting of the plant and of the supervisory controller. Both the open loop model and the specifications can be stated in terms of automata—a state space approach—or in terms of languages—an input-output approach. Just as in classical control theory concepts like controllability were introduced by the ‘‘technology’’ of distinguishing controllable and uncontrollable events (variables). Once this supervisory control theory paradigm was accepted, it became possible to parallel many of the developments of classical control theory. Optimality of a controller can be expressed via ‘‘maximal permissiveness’’.

While the link to systems theory is obvious, and important, there are also suggestive links to computer science. The concepts of controllability and non-blockingness had been studied by computer scientists under the names of safety and liveness. Bisimulation (Milner, 1980) is related to the minimal plant model, and to observability.

Why was this rapid progress possible? The framework is highly flexible with respect to the choice of models. The state representation can be totally unstructured as in automata, or it can be structured as vector integer or vector Boolean, or it can be any combination thereof. The theory is independent of the representation; it is independent of the implementation technology as well. It allows modularity, leading to decentralized and hierarchical technology. And last but not least it leads to constructive approaches to control synthesis, which can be embodied in software packages. Thanks to this flexibility it was relatively easy and fast to extend the theory from the initially purely logical dynamical systems, to real-time supervisory control, including also forced events and hard real-time specifications; these extensions turned out to be fairly straightforward.

Whether these developments have already led to many commercial applications is hard to judge. Certainly there are several industrial centers that are known to develop and package the engineering use of supervisory control theory. But some of these developments are proprietary, and hence are not known very well to the outside world. There is, however, a wide range of potential applications that have been discussed in many papers. The bottleneck is that real applications work with very large state spaces. How to handle this challenge is a topic that is obviously of great concern for future development, to be discussed below.

For Guy Cohen, and other members of the MAXPLUS group that introduced the DES ideas completely independently, it started around the same time, in August 1981. They tried to ‘‘surf on the then fashionable wave’’ of flexible manufacturing systems. They observed that the usual operations research approach based on queueing theory and Poisson randomness was not really relevant to control of flexible manufacturing. The execution times could be modeled much better as deterministic variables rather than as exponentially distributed random variables, and many interesting phenomena depended on periodic behavior, rather than on ergodic behavior. These periodic regimes could be observed from very simple ''manual'' simulations of FMS. While trying to explain these periodicities they observed that Gantt charts could be expressed using the $(\max, +)$-linear algebra that others had studied before as a purely mathematical object. Their model of an FMS was based on very simple ''first principles''. An operation in a manufacturing system can start only if the required parts have been finished upstream. And they ''closed the loop'' by assuming that the next ''raw part'' can enter the line of machines only after a finished part has left the production line.

This observation was the starting point for a fruitful interplay between systems theory, mathematics, and applications (Cohen et al., 1991). It turns out that many dynamical systems that were studied in operations research could be modeled as linear dynamical systems, by using $(\max, +)$ representations. The periodic regime was nothing but the expression of an eigenvalue/eigenvector problem in an unusual linear dynamic system, one where addition is replaced by maximization and multiplication by addition. The dynamic behavior of these systems, which looked very complicated at first, could be treated as a linear system. While the relation between manufacturing systems and $(\max, +)$ algebras had been observed by Cuninghame-Green (1979) 20 years earlier, it was only the interpretation of this relation as a linear dynamical system that allowed its successful use. As so often in mathematics and in systems theory, success depended on using the appropriate mathematical notation, and on being acquainted with the right system-theoretic framework.

Of course the $(\max, +)$ theory provides a linear description only for a limited class of plants. But just as classical linear control theory has been hugely successful despite the fact that all plants are non-linear to some extent, one can also hope to apply the theory to a much wider class of systems. Progress in this direction is undoubtedly necessary in order to make the theory sufficiently mature for significant applications in the field.

Using this system-theoretic paradigm the MAXPLUS group and others developed a linear systems theory, dealing with transfer functions and the frequency domain approach to $(\max, +)$ systems, and developing controllability and observability concepts related to linear feedback control (Cohen et al., 1999). Just like Ramadge and Wonham (1987) put supervisory control theory, based on automata representations of plants, in the framework of systems theory, so did $(\max, +)$ theory revive another class of dynamic plant models by putting them in the dynamical systems theory framework. The viewpoints of supervisory control theory and $(\max, +)$ theory seem very different at first sight. The first is based on logical expressions of specifications, while the second is more oriented towards quantitative performance evaluation of real-time systems. But both are definitely control system theory based. And both use similar algebraic and lattice theoretic structures in the development of their algorithms.

While the generality of the supervisory control paradigm makes it extremely flexible and adaptable to many different applications, a choice of state space representations is necessary when explicitly writing down an algorithm. Alessandro Giua discussed one particularly useful class of representations, based on Petri nets. He pointed out that 40% of the papers in WODES2000 were related to Petri nets. The reason for this success is probably that the state space of a Petri net is structured as a vector, a concept very familiar to a control engineer.

Unlike other formalisms in DES Control theory, Petri nets are not exclusively used—not even mainly used—in our control community. There exists a well established ''Petri net community'', mainly consisting of computer scientists. This has the advantage that we can make use of a wide range of methods and tools developed by these ''Petri netters''. But this overlap also has a disadvantage. Often it is felt that the Petri net community is too closed, and that many of the main results on Petri nets are part of a folklore, which is not written down in a form that is accessible to the outside world. There are few reference books, nor are there any journals bringing together the Petri net results. In any case very few of the Petri net publications have a systems theoretic flavor. All too often the vector structure of the state space is hidden behind a graph notation.

The success of Petri nets for DES control can be attributed to the fact that there is a large ''family of Petri net models'' crossing many boundaries. Many different types of discrete event dynamical systems are represented by different types of Petri nets. Place/transition nets represent purely logical DES and are useful for supervisory control theory. Deterministic timed event graphs (a subclass of Petri nets) are equivalent to $(\max, +)$-linear systems discussed by Guy Cohen. More general timed deterministic and stochastic Petri nets can be used for analysis and performance evaluation of DES systems involving real time specifications. High level nets and hybrid nets can be used for representing hybrid dynamical systems. Generalized stochastic Petri nets can model general Markovian models, useful in stochastic optimization problems as discussed below, by Xi-Ren Cao. All these Petri net models can be used for simulation, for analysis and for control synthesis, for formal verification and for optimization.

A second boundary crossed by Petri net models is that deriving from the different stages in the design of a control system. Petri nets have been used for all phases starting from plant modelling and the specification of a control goal, via analysis and optimization, and finally for implementation. The GRAFCET formalism that is often used in the implementation of PLCs is indeed very closely related to Petri net representations.

The research activity using Petri nets for DES control has been very fruitful over the last 10 years. In order to illustrate this, Alessandro Giua discussed some applications to supervisory control (Holloway et al., 1997). Some attempts at representing DES plants with a logical control closed loop were made in the early days of Petri nets. But these attempts were not very successful because there was no clear separation of plant and controller, and there was as a result no clear methodology for implementing the controller. The situation changed after the introduction of the supervisory control paradigm by Ramadge and Wonham (1987). Most of the research on Petri nets for discrete event dynamic systems in the 1990s was strongly influenced by this supervisory control paradigm.

One formulation of the Petri net control problem uses event feedback. The supervisor observes events generated by the plant and blocks some controllable events with the aim of avoiding a set of forbidden event sequences. This is still an open area, where progress is limited so far, because the Petri net languages do not have a structure that is algebraically simpler than the languages for any other automaton.

The second approach is based on state feedback. The supervisor observes the state of the plant—represented by the marking of the Petri net model—and blocks controllable transitions with the aim of avoiding ''forbidden'' states. The structure of Petri nets is particularly well suited to this problem, and significant progress in this direction has been made over the last ten years. The results obtained so far deal mainly with safeness of the plant, with avoiding dangerous conditions. However, good supervisory controllers must also deal with ‘‘liveness’’ (which is equivalent to the non-blocking properties in supervisory control theory).

The last panel member to present his view on past and future developments was Xi-Ren Cao, who presented the performance optimization point of view on discrete event dynamical systems. Based on seminal work by Larry Ho various techniques, like perturbation analysis, sensitivity analysis and ordinal optimization, have been developed for synthesizing optimal controllers for DES. The goal was to estimate gradients and hence to optimize a system, based on the observation of one single sample path (Ho, 1987; Ho et al., 1983). Just like the flexible manufacturing systems that served as motivation for the $(\max, +)$ paradigm presented by Guy Cohen, the work on perturbation analysis was also motivated by automated production lines. In fact the perturbation analysis work was stimulated by a real world problem, a project on an ‘‘automated line monitoring system’’.

These methods exploit the structure of the timed automata in order to estimate derivatives—sensitivity measures—of the performance of a discrete event system, just like dual variables provide ‘‘local linearization’’ information on the change in cost in maximum principle type results. In a more general setting, sample path based analysis can be considered in the context of ‘‘learning from experience’’. Markov chain Monte Carlo, reinforcement learning, neural network training, etc. are all examples. In fact without specific structural assumptions, such as linear differential equations, system performance must be learned from simulations or from observed sample paths, using all possible control values at least once. Using structural information on the DES under study it is possible to synthesize optimal controllers from one single trajectory, without the need to repeat experiments for every control value. This makes the approach applicable for on-line control of real engineering applications. It also reduces computational efforts.

This new mind set for the engineering community, starting from the initial work on perturbation analysis, has led to significant theoretical work on simulation and on optimization. It has demonstrated the ability to obtain sensitivity information merely by observing the operation of a complex system, and of inferring its perturbed behavior via simple nonintrusive means.

Perturbation analysis has also led to significant applied research in manufacturing systems (scheduling, priority queueing, optimal number of Kanbans), in communication networks (capacity and flow control, admission control), and in air traffic control. Some real world applications have been highly successful. Perturbation analysis has been applied to fine-tuning of a scheduler for an elevator system (Pepyne and Cassandras, 1998). The fact that perturbation analysis uses structural information on the plant led to a learning time of five days, as compared to one year for the AI competitors using neural networks.

The research discussed by Xi-Ren Cao emphasized the interrelation between discrete event systems, and performance optimization. As such it closed the loop, by re-iterating the fact that the DES work of the systems community has allowed a new view on the dynamic programming tools traditionally used in the OR community. One example of the new insights the state based systems paradigm leads to is the use of Markov potential theory (Cao, 1998), which shows that the traditional Markov decision problem (MDP) can be viewed as the performance sensitivity problem in a discrete policy space.

The Near Future

More applications of perturbation analysis, and of the other discrete event dynamical system paradigms, are yet to come. Besides further progress on the theoretical side, this will also require overcoming a common barrier in any technology transfer process: much more effort is needed in order to make the application of the new paradigms in industry simpler and easier. Most methods are very strongly problem dependent, discouraging cooperation between industry and academia. As pointed out in comments from the audience, a lot more effort in education is also needed in order to bridge this industry-academia gap.

Xi-Ren Cao proposed that for the perturbation analysis paradigm, the following three important and challenging topics should be singled out for further research. Most applications require mixed modeling, with some components modeled as continuous systems, other components discrete. Sample path based methods for on-line optimization of hybrid systems would be extremely useful; both the Markov potential approach and the finite perturbation analysis may provide a good base. Extensions of single sample path based optimization to on-line optimization of distributed Markovian systems, and partially observed Markovian systems, are also feasible. The ultimate task on the road to practical applications also includes development of generally applicable and at the same time computationally efficient algorithms.

Alessandro Giua also identified hybrid systems modeling, analysis and control, as an emerging and promising area where extensions, belonging to the Petri net family, can be useful. Another challenge in control to the Petri net community is the combination of safeness and liveness specifications. A practically useful controller will not only achieve the maximally permissive safe operation of the plant, but it should also ensure that something ‘‘good’’ will eventually occur. Problems of this kind include deadlock avoidance, fairness, and liveness in the Petri net sense. The development of effective synthesis tools achieving these combined goals is a hot topic for the coming years. Several papers at WODES2000 (Boel and Stremersch, 2000) dealt with this topic.

The discussion returned several times to the proper representation of discrete event dynamical systems via different types of Petri nets. Alessandro Giua pointed out that high level nets (e.g., colored Petri nets) have been used successfully for modeling when there is a lot of symmetry in the plant model. It is often not clear though how this symmetry carries over to control. The importance of a proper structure for the state space representation was reiterated several times, including the remark that the structure should not only be suitable for modeling but also for developing feedback controllers.

In a related response, Murray Wonham pointed out that it took more than 20 years before Petri nets were recognized as dynamical system models by the systems community. Why is this so? Petri nets would be a lot easier to understand, and would have gained faster acceptance by system theorists, if they would have been expressed as recursive, dynamic systems, not as graphs.

The panel members did not agree on whether automata or Petri nets are the most suitable as the ''lingua franca'' for discrete event dynamical systems. In fact there was an agreement in the end that, while dynamical systems theory is a common framework, it is too early to have any standardization of representations or models. Our field is still too young for limiting our view to (or by) one single paradigm. The development of DES illustrates that finding the right ''language'' to talk about something is crucial in solving a problem.

Guy Cohen also pointed out that good application can only come about by a better theoretical understanding of the structure of discrete event dynamical systems. For the (max, +) paradigm an important theoretical development that is needed is the integration of algebraic and geometric tools for control synthesis. For classical linear systems Murray Wonham has shown how the use of geometric concepts such as controllability and observability subspaces, closely related to classical linear agebra, can be fruitful for control synthesis. For the (max, +) algebra a similar duality must probably be introduced. Idempotent semimodules are for the (max, +) algebra what vector spaces are for classical linear algebra. Even some convexity concepts may be close at hand in these idempotent semimodules, and convexity is one way towards differential calculus (through subdifferential calculus). Such tools may equip us for further exciting developments in the story of discrete event systems.

Murray Wonham concluded that there is no lack of excellent things to do, both in terms of developing applications and in terms of developing the theory necessary to facilitate these applications. He also agreed with other speakers that further applications require that supervisory control theory be made friendly and accessible to potential industrial users. Layering and zooming may make the ''display'' much more user friendly. Among the application areas where Murray Wonham knows of proposed applications are manufacturing, robotics, as well as chemical process control, protocol design in communications networks. Feature interaction management in telephony and fault diagnosis are also important current developments.

One area where discrete event systems are currently used is the design of PLCs. The question was raised what extensions should be made to the theory in order to be able to handle distributed control systems (DCS) as well. The area of decentralized control systems is clearly very important. Murray Wonham pointed out that this may again be related to the dilemma between generally applicable flat structures that require little domain-specific knowledge versus computationally more efficient layered structures for the (very large state space) of the discrete event models.

The size of the state space is a major hurdle to overcome. Decentralization, and other forms of ''system architecture'' such as networked control, may be one way to tackle this. The trade-off between efficiency and implementability, between convenience and computational efficiency, must be considered when studying decentralized control systems.

Building the bridge between theory and applications will require progress in two main areas. Quantitative performance evaluation must be combined with discrete event models, perhaps via stochastic ''overlays'' over discrete event models (compare this to the future work proposed by Xi-Ren Cao). Mixing (max, +) with probabilistic models is useful, and many interesting results in this direction can be found in Cohen et al. (1991). And the hybrid connection must be made, combining discrete event models with continuous flow models, especially as a bridge to fault detection applications.

These developments require the solution of several interesting theoretical challenges. The important epistemological question underlying all of discrete event systems theory is how to understand complex system. Managing the exponential state space explosion for complex systems requires improved architectural approaches for structuring the discrete event models. Exploiting mixed state models with state spaces that are Cartesian products of unstructured states of automata components, combined with vectors of integer or Boolean variables will prove useful. But just as was mentioned in the discussion on the decentralized architecture it is very important that one properly understands the trade-off between convenience and computational complexity in the general case as well.

These are difficult problems, but there are grounds for optimism here. These state space structures are related to canonical functional representations through ‘‘universal’’ (categorical) constructions. And at the same time their mathematical base is elementary, uncluttered by extraneous technicalities (in DES we never have to deal with nasty singularities as occur in non-linear systems theory). Ultimately we will be able to understand and to manage large, complex systems.

Via Education and Tools to Real Applications

Both the panel members and the interveners from the floor emphasized repeatedly that education and tools are the two most pressing needs on the way to real applications. Some panel members felt that it would be very difficult to establish an undergraduate course on discrete event systems, either because the subject was too advanced or because it would be perceived by most students and colleagues as too specialized. They felt that it was important to integrate DES topics in general system theory courses (or queueing theory courses), emphasizing the basic dynamic systems concepts.

Others, however, like Murray Wonham, pointed out that the undergraduates in engineering these days have a very good knowledge of discrete mathematics, and often have enough algebraic maturity to appreciate the contents of such a discrete event dynamical systems course. It is a topic that allows interesting laboratory work, designing for example agent based controllers. In fact this approach has been successfully implemented in a freshman course in Berkeley (Lee and Varaiya, 2000).

Guy Cohen also pointed out that one can develop a very interesting general linear system theory course with all the common basic concepts. This course can then encompass as different ‘‘cases’’ physical linear systems as well as timed event graphs, by selecting different semantics for the $+$ and $\times$ symbols.

Bengt Lennartsson explained that at Chalmers University they have ten years of experience with a DES course in a combined automation/mechatronics curriculum for electrical, mechanical and computer science students. Students there get theoretical and applied education including PLCs, supervisory control theory, Petri nets. The experience is that programming PLCs even when it is based on Grafcets is easy, just as state based feedback control is easy to explain. But language based subjects and the relation between supervisory control theory and PLCs are seen as very hard by the students.

This course at Chalmers Univeristy uses a software tool DESCO that has been explained in the tools session at WODES2000 (Fabian and Hellgren, 2000). The availability of (or rather the lack of) easy to use tools—preferably with a graphical interface—was felt to be a major problem for establishing a discrete event dynamical systems course (whether at the undergraduate or the graduate level). Students must be able to run interesting examples. No commonly accepted tool for this purpose, like MATLAB for classical dynamical systems, currently exists. Besides DESCO, some other tools have been developed specifically for teaching purposes and are made available by their developers for general use (like UMDES [UMDES] at University of Michigan, and CTCT at the University of Toronto [UTOR]). Several other tools for specific applications were demonstrated at the tools session of WODES2000 (Boel and Stremensch, 2000). But it is unrealistic to expect that these freely available tools can be maintained at the same level as a commercial tool like MATLAB. And none of these tools has yet found general acceptance.

Easily used tools with nice graphical interfaces, perhaps more powerful and with more emphasis on computational efficiency than the tools for education, are also needed in order to allow industry to develop real applications of discrete event dynamical system theory. Some proprietary in-company development is currently going on. At WODES2000 Bertil Brandin discussed the VALID tool developed by Siemens. Other companies like Rockwell are also developing tools, but these are not made available for outside use.

It would be useful to build up a comprehensive data base of tools, whether for teaching or for industrial use, whether freely available, shareware or commercially sold, whether professionally maintained or not. Felisa Vazquez-Abad explained that INFORMS maintains a web site with simulation tools, that is consulted quite a lot. Of course it took a significant amount of work to establish such a web site, but by limiting the web site to linking up with pages of other researchers, the web site development could be kept light weight, with a small number of pages. Nevertheless the INFORMS web site does require a professional webmaster in order to maintain it as a useful tool.

According to Alessandro Giua one other important source for education could be a textbook on Petri nets, that explains the family of Petri net models within the broad context of dynamical systems theory. Currently the best systems theoretic discussions of Petri nets, accessible to control theorists, can be found in Cassandras and Lafortune (1999) and Moody and Antsaklis (1998).

Slavek Bulach pointed out that applications do not necessarily require software development only. It has been demonstrated that Petri net controllers can be built in hardware as well. This use of ASICs makes the computational speed much higher, and allows more demanding, time critical applications.

Conclusions: Unity in Diversity, Diversity in Unity

Discrete event systems theory has many parents; it has grown out of systems theory, computer science, real-time control; manufacturing systems and OR have also had an impact. As a result many different paradigms are used to describe discrete event systems. In order to make further progress in applying DES control theory it is clear that practitioners and theoreticians alike must understand each other's language, and the many dialects that separate theoreticians from each other. It is unlikely that in the foreseeable future a common ‘‘Esperanto’’ will be created. Nor is this desirable. The field is too young for standardization. Researchers should continue exploring many different avenues, in different paradigms or languages, both when developing new theory and for applications. Only by doing this can we find out what the ‘‘winning paradigms’’ will be.

But at the same time we should be aware that many common features do abound in the different paradigms in our field. The lattice theoretic flavor common to both supervisory control theory and to residuation theory for $(\max, +)$ systems is a good example. When developing courses and tools (both for education and for industrial applications) it is important to recognize these common features, and to deal with them explicitly in a way that is understandable by those using a different language.

Conferences like WODES bringing together people speaking these different languages will continue to force us to speak these languages in such a way that colleagues from other ‘‘languages groups’’ can understand us. Diversity enriches science (as well as society) provided we show enough respect for each other, and genuinely try to understand the points of view expressed in the others’ language. As Xi-Ren Cao expressed it, ‘‘we don’t need to all talk the same language, but we should cooperate across languages’’, as in a multicultural society.

Progress also depends on studying good applications. Communications systems, transportation, and manufacturing were mentioned as important fields with good applications. But good applications can only come about if we have well educated students, if good tools are developed, and if good theory is available.

Progress in the field of DES control will require progress in each of the subfields mentioned; this progress will also require a better understanding of their interrelationship. In all these respects we can conclude with Murray Wonham that ‘‘there is no lack of excellent things to do’’.

The WODES umbrella provides a good place for developing this ‘‘diversity in unity’’. From the wide range of paradigms researchers should distill general principles, in order to achieve ‘‘unity in diversity’’. Let us hope that at future WODES meetings we will see the results of this progress, and that the DES community will be able to talk together about new challenges bringing theory even closer to real applications.

References

Boel, R., and Stremersch, G. (editors) 2000. *Discrete Event Systems: Analysis and Control*. Kluwer.

Cao, X.-R. 1998. The relation among potentials, perturbation analysis, Markov decision processes, and other topics. *Journal of Discrete Event Dynamic Systems* 8: 71–87.

Cassandras, C., and Lafortune, S. 1999. *Introduction to Discrete Event Systems*. 2nd ed., Kluwer.

Cuninghame-Green, R. 1979. *Minimax Algebras*. Berlin: Springer Verlag.

Cohen, G., Gaubert, S., and Quadrat, J.-P. 1999. Max-plus algebra and systems theory: where we are and where to go now. *Annual Reviews in Control (Elsevier-IFAC)* 23(1): 207–219.

Cohen, G., Moller, P., Quadrat, J.-P., and Viot, M. 1991. Algebraic tools for performance evauation in discrete event systems. In *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World* (Yu-Chi Ho, ed.), IEEE Press.

Fabian, M., and Hellgren, A. 2000. Desco—a tool for education and control of discrete event systems. In *Discrete Event Systems: Analysis and Control* (R. Boel and G. Stremersch, eds), Kluwer, pp. 471–472.

Ho, Y.-C., Cao, X.-R., and Cassandras, C. 1983. Infinitesimal and finite perturbation analysis for queueing networks. *Automatica* 19: 439–445.

Ho, Y.-C. (editor) 1991. *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World* IEEE Press.

Ho, Y.-C. Discrete event dynamic systems—perspectives and open problems. *IEEE-T-AC-32*: 563–572.

Holloway, L., Krogh, B., and Giua, A. 1997. A survey of petri net methods for controlled discrete event systems. *Journal of Discrete Event Dynamic Systems: Theory and Applications* 7(2): 151–190.

Kalman, R., Arbib, M., and Falb, P. 1969. *Topics in Mathematical Systems Theory.* McGraw Hill.

Lee, E., and Varaiya, P. 2000. Introducing signals and systems—The Berkeley approach. *Proceedings of the First Signal Processing Education Workshop*. Hunt: Texas, Oct. 15–18, see also: http://ptolemy.eecs.berkeley.edu/ ∼ eal/eecs20/

Moody, J. O., and Antsaklis, P. 1998. *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer Academic Publishers.

Milner, R. 1980. *A Calculus of Communicating Systems, Lecture Notes in Computer Science* 92, Springer Verlag.

Pepyne, D., and Cassandras, C. 1998. Design and implementation of an adaptive dispatching controller for elevator systems during uppeak traffic. *IEEE-T-CST* 6(5): 635–650.

Ramadge, P., and Wonham, W. M. 1987. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization* 25(1): 206–230.

UMDES http://www.eecs.umich.edu/umdes

UTOR CTCT software (v. 2001.07.01), and notes on control of discrete event systems (rev. 2001.07.01). Available online at http://www.control.utoronto/people/profs/wonham/wonham.html