Predicting the Functional/Performance Impact of Dynamic Power Management *

M. Bernardo, A. Bogliolo, A. Acquaviva, A. Aldini, E. Bontà, E. Lattanzi
Università di Urbino "Carlo Bo"
Istituto di Scienze e Tecnologie dell'Informazione
Piazza della Repubblica 13, 61029 Urbino, Italy

1. Power-Aware Computing

Reducing the power consumption is a fundamental criterion in the design of battery-powered devices typical of modern mobile embedded systems. Significant power savings can be achieved at run time through the application of dynamic power management (DPM) techniques [3], i.e. techniques that – based on run time conditions – modify the power consumption of the devices by changing their state or by scaling their voltage or frequency.

Electronic systems are designed to deliver peak performance, but they spend most of their time executing tasks that do not require such a performance level. For instance, hand-held personal digital assistants are mainly used to run interactive applications (such as personal organizers and text editors) whose main task is capturing sparse input events, while cellular phones are reactive systems that are usually idle waiting for incoming calls or user commands.

In general, electronic systems are subject to time-varying workloads. Since there is a close relation between power consumption and performance, the capability of tuning at run time the performance of a system to its workload provides great opportunity to save power. DPM techniques dynamically reconfigure an electronic system by changing its operating mode and by turning its components on and off in order to provide at any time the minimum performance/functionality required by the workload while consuming the minimum amount of power.

The application of DPM techniques requires: (i) power-manageable components providing multiple operating modes, (ii) a power manager having run-time control of the operating mode of the power-manageable components, and (iii) a DPM policy specifying the control rules to be implemented by the power manager. The simplest example of power-manageable hardware is a device that can be dynamically turned on and off by a power manager that issues shutdown and wakeup commands according to a given policy. When turned on, the device is active and provides a given performance at the cost of a given power consumption. When turned off, the device is inactive, hence provides no performance and consumes no power.

In all cases of practical interest, however, shutdown and wakeup transitions have non-negligible costs both in terms of energy and in terms of time. Transition costs make the design of DPM policies a non-trivial task for two main reasons. First, a shutdown can be counterproductive if the idle period is not long enough to compensate for the transition energy. Second, if a service request is issued when the device is inactive, the wakeup time adds a delay to the service time that may cause an unacceptable performance degradation. In practice, transition costs limit the actual exploitability of low power states and make it necessary to predict user idleness to take DPM decisions.

The DPM techniques can be classified on the basis of: (i) the predictor they use, (ii) the degree of control granted to the power manager, and (iii) the nature of the decisions it takes.

Existing predictors differ from each other both for the target of the prediction and for the observed history used to make predictions. As far as the exploitation of an inactive state is concerned, the prediction target may be either the occurrence probability of idle periods longer than the break-even time, or the expected length of the next idle period. Similarly, predictions may be based either on the average length of the last n idle periods, or on the length of the last activity burst, or on the first part of the current idle period.

Depending on the degree of control that the power manager has on the system, we distinguish between two main classes of DPM techniques. We call shutdown techniques those in which the power manager can only trigger shutdown transitions, while wakeup transitions are triggered by incoming requests. We call preemptive techniques those in which the power manager may issue both shutdown and wakeup commands and tries to preemptively wake up the system in order to reduce performance penalties.

Finally, we distinguish between deterministic and

* Co-financed by Regione Marche within the CIPE 36/2002 framework.

stochastic DPM policies. Deterministic policies take deterministic decisions based on the observed working conditions: the same decision is taken whenever the same conditions occur. Stochastic policies take randomized decisions whose probabilities depend on the observed working conditions: different decisions may be taken under the same conditions.

2. Impact-Predicting Methodology

Whatever policy is adopted, the introduction of the DPM within a mobile computing device may have a non-negligible impact on the overall system functionality and performance. It is therefore of paramount importance to assess such an impact before the DPM is introduced, in order to make sure that the system behavior will not be significantly altered and that an intolerable degradation of the quality of service will not occur.

This objective can be achieved by following a methodology that helps predicting the effect of the DPM through the comparison of the functional and performance characteristics of the system without and with DPM. Since it should be applied in the early stages of the system design – in which a high level of abstraction is admitted that favors the task of verifying properties – the predictive methodology can take advantage of formal description techniques, like e.g. stochastic process algebras [4] and stochastic Petri nets [2], as well as formal analysis techniques, like e.g. equivalence checking [6] and model checking [5].

Although formal methods have already been successfully applied to the specific tasks of optimizing DPM policies [10, 12] or reasoning about power-constrained real-time systems [13], they can be exploited within a broader setting in which the objective is to predict whether the adoption of a specific DPM policy is convenient or not, by incrementally investigating the functional and performance transparency of the considered policy. Obviously the impact-predicting methodology can also serve for more specific purposes, because it can be used to tune the DPM operation parameters in order to achieve a satisfactory energy-quality tradeoff (if any).

For the sake of the application of the methodology, it is useful to divide the DPM activities into two classes. The activities of the first class are the ones that modify the state of the power-manageable device, while the activities of the second class are the ones that collect information about the state of the power-manageable device. When the DPM is capable of modifying the state of the power-manageable device – in which case the behavior and the efficiency of the overall system may be altered – we say that the DPM is enabled. On the contrary, when the state-modifying activities

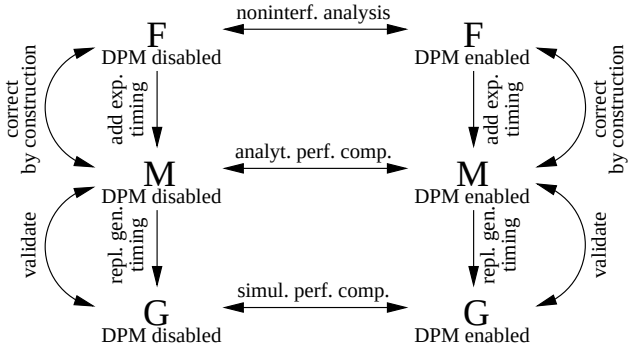

Figure 1. Models and phases of the impact-predicting methodology

of the DPM cannot be performed, we say that the DPM is disabled.

The methodology to predict the effect of the DPM on the functionality and the performance of a mobile battery-powered computing device, which we recall from [1], is shown in Fig. 1. As can be seen, the methodology requires to build three pairs of models of the system – functional, Markovian and general – each of which is incrementally obtained from the previous one by adding further details. Within each pair, one model refers to the system with the DPM disabled, while the other one refers to the system with the DPM enabled. The functional, Markovian and general models are then compared two by two in three different phases, in order to investigate the DPM impact on the system functionality and performance.

2.1. Noninterference Analysis of the Functional Models

The two models considered in the first phase of the methodology address only the behavior of the system. These models are used to check whether the introduction of the DPM alters the system functionality or not.

In order to assess the functional transparency of the DPM, the methodology resorts to the noninterference analysis approach [9]. The general idea behind this approach is to view a system execution as an information flow and to consider that a group of system users (high users), employing a certain set of commands, is not interfering with another group of system users (low users) if what the first group of users can do with those commands has no effect on what the second group of users can see. This approach has traditionally been applied for security purposes, as noninterference analysis can reveal direct and indirect information flows that violate the access policies based on assigning different access clearances to different user groups.

In the framework of the DPM-related methodology, the

noninterference analysis is used to check whether the DPM interferes with the behavior of the system as observed by the system clients. In fact, from the noninterference perspective, the state-modifying activities carried out by the DPM are the only high ones, whereas all the activities carried out by the system clients are the only low ones.

The predictive methodology adopts the version of the noninterference analysis approach based on equivalence checking [8]. Establishing noninterference thus amounts to verifying whether – from the client standpoint – the functional model of the system with the state-modifying activities of the DPM being made unobservable is equivalent to the functional model of the system with the same activities being prevented from taking place (i.e. with the DPM disabled). In the verification process, all the activities that are classified as being neither high nor low have to be made unobservable as well.

The formal notion of equivalence that is employed to carry out this task is weak bisimulation [11], which relates two system models whenever they are able to mimic each other's behavior while abstracting from unobservable activities. Should the two functional models above turn out not to be weakly bisimilar, a Hennessy-Milner logic formula can be automatically obtained that distinguishes the two models, thereby explaining why they are not equivalent [7]. This formula can then be used as a diagnostic piece of information to guide the modification of the behavior of the DPM and/or the system in order to achieve noninterference.

It is worth pointing out that establishing noninterference prior to the introduction of a specific timing of the system activities, which may rule out some behaviors, ensures that the DPM is functionally transparent in itself, not because of the adoption of particular assumptions.

2.2. Analytical Performance Comparison of the Markovian Models

Once functional transparency has been achieved, it has to be investigated whether the DPM affects the system performance. While in general the impact of the DPM on the system behavior can be avoided by means of suitable modifications, it is practically impossible that the introduction of the DPM does not alter the quality of the service delivered by the system. Therefore, the purpose of this further investigation is to find a balance between the power consumption and the overall system efficiency.

In the second phase of the methodology the two functional models are made more complete by specifying the timing of each system activity, thus allowing for performance evaluation. Since the activity durations are expressed in this phase through exponentially distributed random variables, the derived models are Markovian models yielding continuous-time Markov chains.

These models do not need to be validated against the corresponding functional models, since they are directly obtained from the latter by attaching exponential delays to the state transitions. In other words, the two Markovian models are consistent by construction with the corresponding functional models, in the sense that the state space of each of the two Markovian models is isomorphic (up to the transition delays) to the state space of the corresponding functional model. As a consequence, whenever the two functional models meet noninterference, then so do their corresponding Markovian models.

When applying the methodology in practice, the correctness by construction and the resulting preservation of noninterference depend on the precise way in which the functional models are extended as well as on the expressive power of the formalism adopted to develop the models. As an example, in order to measure the percentage of time that the system spends in states characterized by different power consumption levels, it may be necessary to introduce self-looping transitions (with arbitrary exponential delays) in the considered states. Since they are neither high nor low, hence they can be made unobservable, such additional transitions do not affect noninterference.

More troublesome can be the specification of the fact that the duration of certain activities is negligible from the performance viewpoint, which is accomplished through the so called immediate transitions provided by some formalisms (see, e.g., [2, 4]). Since the immediate transitions take precedence over the exponentially timed ones, their use may alter the state space of the Markovian models with respect to the state space of the corresponding functional models. As a consequence, noninterference may not be preserved. Assuming that the immediate transitions are consistently used in the two Markovian models, the noninterference analysis should be repeated in the second phase only if some of the state-modifying activities of the DPM are characterized through immediate transitions. The reason is that, since these activities are the only ones to be enabled in one model and disabled in the other model, they are the only source of potential violation of weak bisimulation when they take precedence over other activities.

The two Markovian models can be solved analytically through standard techniques. This opens the way to the comparison of the system with the DPM disabled and with the DPM enabled on the basis of certain performance measures – like power consumption, system throughput, radio channel utilization, and quality of service – obtained when varying the DPM operation rates. Such performance indices can easily be expressed through a combined use of cumulative and instantaneous rewards. This investigation can then be exploited to tune the frequency of the DPM operations, in such a way that a reasonable tradeoff between the power consumption and the overall system efficiency is achieved.

2.3. Simulative Performance Comparison of the General Models

In the third phase of the methodology the two Markovian models are made more realistic by replacing the exponential distributions with general distributions wherever necessary to better characterize the actual delays.

Since substituting general distributions for exponential distributions may not be a smooth process, the general models may need to be validated against the corresponding Markovian models. For instance, in those formalisms that do not directly support general distributions (see, e.g., [4]), major modifications of the Markovian models are needed in addition to the distribution replacement. In such a case it is necessary to assess the consistency of each of the two general models with respect to the corresponding Markovian model. This is accomplished by verifying that both models result in comparable values for the considered performance measures, when substituting exponential distributions back for general distributions in the general model in a way that preserves their expected values.

As another example, noninterference may not be preserved. In fact, the replacement of exponential distributions with general distributions no longer having infinite support may alter the state space of the general models with respect to the state space of the corresponding Markovian models. This is similar to what happens in the Markovian models when using immediate transitions. Therefore, assuming that the distributions with finite support are consistently used in the two general models, the noninterference analysis should be repeated in the third phase only if some of the state-modifying activities of the DPM are characterized through distributions with finite support.

Once the validation succeeds, the two general models can be simulated via standard techniques in order to estimate at a certain confidence level the same performance measures considered in the second phase with the DPM disabled and with the DPM enabled. The comparison of the resulting figures should then guide the decision about whether it is worth introducing the DPM in certain realistic scenarios. If so, the figures should also help tuning the DPM operation rates without compromising the achievement of the desired level of quality of service.

We conclude by observing that the second and the third phase both refer to a performance comparison of the system with the DPM disabled and with the DPM enabled. Since the third phase addresses more realistic scenarios, one may want to skip the second phase, thus going directly from the first one to the third one. Although possible, this is not recommended. In fact, even though the Markovian models may not be realistic, the performance figures obtained from their analytically derived solution constitute the only means to validate the simulation results of the general models in the early stages of the system design. Moreover, skipping the second phase would introduce a gap in the incremental modeling process enforced by the methodology, which may likely cause inconsistencies between the general models and the corresponding functional models.

References

[1] A. Acquaviva, A. Aldini, M. Bernardo, A. Bogliolo, E. Bontà, and E. Lattanzi, *"A Methodology Based on Formal Methods for Predicting the Impact of Dynamic Power Management"*, in *Formal Methods for Mobile Computing*, LNCS 3465:155-189, 2005.

[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *"Modelling with Generalized Stochastic Petri Nets"*, John Wiley & Sons, 1995.

[3] L. Benini, A. Bogliolo, and G. De Micheli, *"A Survey of Design Techniques for System-Level Dynamic Power Management"*, in IEEE Trans. on VLSI Systems 8:299-316, 2000.

[4] M. Bernardo, L. Donatiello, and P. Ciancarini, *"Stochastic Process Algebra: From an Algebraic Formalism to an Architectural Description Language"*, in *Performance Evaluation of Complex Systems: Techniques and Tools*, LNCS 2459:236-260, 2002.

[5] E.M. Clarke, O. Grumberg, and D.A. Peled, *"Model Checking"*, MIT Press, 1999.

[6] W.R. Cleaveland and O. Sokolsky, *"Equivalence and Preorder Checking for Finite-State Systems"*, in *Handbook of Process Algebra*, Elsevier, pp. 391-424, 2001.

[7] W.R. Cleaveland, *"On Automatically Explaining Bisimulation Inequivalence"*, in Proc. of CAV 1990, LNCS 531:364-372, New Brunswick (NJ), 1990.

[8] R. Focardi and R. Gorrieri, *"A Classification of Security Properties"*, in Journal of Computer Security 3:5-33, 1995.

[9] J.A. Goguen and J. Meseguer, *"Security Policy and Security Models"*, in Proc. of SSP 1982, IEEE-CS Press, pp. 11-20, Oakland (CA), 1982.

[10] R.K. Gupta, S. Irani, and S.K. Shukla, *"Formal Methods for Dynamic Power Management"*, in Proc. of ICCAD 2003, ACM Press, pp. 874-882, San Jose (CA), 2003.

[11] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989.

[12] G. Norman, D. Parker, M. Kwiatkowska, S.K. Shukla, and R.K. Gupta, *"Formal Analysis and Validation of Continuous-Time Markov Chain Based System Level Power Management Strategies"*, in Proc. of HLDVT 2002, IEEE-CS Press, pp. 45-50, Cannes (France), 2002.

[13] O. Sokolsky, A. Philippou, I. Lee, and K. Christou, *"Modeling and Analysis of Power-Aware Systems"*, in Proc. of TACAS 2003, LNCS 2619:409-424, Warsaw (Poland), 2003.