

# Evidence-Based Validation and Improvement of Electronic Health Record Systems

Andy Podgurski

EECS Department  
Case Western Reserve University  
Cleveland, OH

podgurski@case.edu

## ABSTRACT

A program of research on methodology for “evidence-based” validation and improvement of electronic health record systems and related health information systems is proposed. This program would integrate existing ideas from software engineering and health informatics with new techniques for analyzing a combination of usage data, clinical data, and execution data, to support scientifically sound methods for measuring and enhancing the safety and efficacy of EHR systems.

## Categories and Subject Descriptors

D.2 SOFTWARE ENGINEERING: D.2.4 Software/Program Verification – Statistical methods; D.2.7 Distribution, Maintenance, and Enhancement. K.4 COMPUTERS AND SOCIETY: K.4.1 Public Policy Issues – Computer-related health issues; Human safety .

## General Terms

Measurement, Reliability, Verification.

## Keywords

Evidence-based software validation and improvement, electronic health record systems, safety, efficacy.

## 1. INTRODUCTION

It is likely that, in the near future, *electronic health record (EHR) systems* will impact the health care of most Americans. Although only a modest number of health care providers currently have comprehensive EHR systems, a larger number of providers have at least some EHR system functionality in place [1], and these numbers are expected to grow rapidly because of the *Health Information Technology for Economic and Clinical Health (HITECH) Act*, which is part of the *American Reinvestment and Recovery Act (ARRA)* of 2009 [2]. The HITECH Act allocates \$19 billion in federal funding (mainly for incentive payments to providers) to expand the use of health information technology, especially EHR systems. One of its goals is to establish a *national health information network (NHIN)* of interoperable health information systems, to enable necessary exchanges of health information to occur expeditiously.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FoSER 2010, November 7–8, 2010, Santa Fe, New Mexico, USA.

Copyright 2010 ACM 978-1-4503-0427-6/10/11...\$10.00.

Widespread adoption of EHR systems has the potential to substantially improve the quality of health care, through better clinical access to patients’ medical information, reduction of medical errors, provision of clinical decision support, and other capabilities [3]. However, these benefits obviously cannot be realized if EHR system do not function properly. Comprehensive EHR systems are complex, multifunctional systems with a number of safety-critical features. Although they do not directly administer or control treatment to patients (at present), they strongly influence the treatment of many patients, and clinicians depend on them functioning appropriately. Software defects and usability problems in such EHR system functions as information display, *computerized physician order entry (CPOE)*, and *computerized decision support (CDS)* can put many patients at serious risk of inappropriate treatment, and unscheduled system shutdowns can paralyze a health care facility for hours. Although EHR systems may not seem as safety-critical as, say, implantable defibrillators, their overall impact on patient health and safety may be much greater. Moreover, the functionality and complexity of EHR systems is likely to increase significantly, e.g., as they incorporate CDS features specifically to support genomic medicine [4].

The development and maintenance of EHR systems poses special challenges to software engineers, which necessitate deviations from standard software engineering practices. Most of these arise from the extent to which the proper functioning of an EHR system must be judged in terms of its *impact on patient outcomes* rather than its conformance to a specification. The fitness for use of an EHR system and even its actual requirements are *emergent properties* that cannot be fully understood, let alone optimized, without observing the interactions between the system, on one hand, and clinicians, patients, and health-care environments, on the other hand.

The challenges posed by EHR systems have precedents in software engineering, but they necessitate a significant realignment in priorities, both for software development and for research. Clearly patient welfare must be the highest priority, well ahead of reducing the costs of development and maintenance and of shortening time-to-market. Accordingly, EHR systems should be produced using the best known development practices for safety critical systems and software [5, 6]. This is not sufficient, however. Much greater priority must be placed on collecting and analyzing data from deployed systems in order to empirically characterize their fitness for widespread use and to improve them when appropriate. This paper will focus on research issues involved in such “evidence based” validation and improvement of comprehensive EHR systems and related health information systems.

Software researchers have placed much emphasis on formal methods of validating software (with respect to a specification) prior to its deployment. Some researchers have even argued that system safety, which they equate with “ultra-reliability” (failure rate lower than  $10^{-7}$  failures/hour), cannot be measured [7]. This position would puzzle medical researchers, who rely primarily on empirical methods and statistical inference in investigating life-critical treatments. The U.S. Food and Drug Administration (FDA) requires vendors of life-critical drugs and devices to evaluate them for *safety* and *efficacy* by conducting clinical trials and by subjecting the results to rigorous statistical analysis [8].<sup>1</sup> On the other hand, FDA does not require that medical devices have ultra-low failure rates, presumably in recognition of the fact that many medical procedures much higher rates of adverse outcomes. For example, a recent international study of non-cardiac surgeries found the death rate to be 1.5% and the inpatient complication rate to be 11% [9].

Formal methods may be valuable in preparing medical software for deployment, but they do not eliminate the need for empirical evaluation of the software under representative operating conditions. Similarly, *synthetic testing*, by which we mean use of any software testing technique that selects or generate test inputs without accounting for the nature of operational system-usage, is not sufficient to establish the safety and efficacy of medical software.

The outline of the remainder of this paper is as follows. Section 2 relates the medical notions of safety and efficacy to the concepts of software correctness and reliability; Section 3 discusses estimation of safety and efficacy measures; Section 4 introduces the ideas of evidence-based validation and improvement of EHR systems; and Section 5 surveys pertinent research directions.

## 2. SAFETY AND EFFICACY VERSUS RELIABILITY

The standard of safety and efficacy for medical interventions is somewhat different from the notions of fitness for use typically applied to software, namely “correctness” and “reliability”. (Security is also important for EHR systems [10], but will not be addressed in this short paper.)

Software researchers have long sought to devise means of ensuring that software is *correct* with respect to its specification, in the sense of producing the specified output for each possible input [11]. However, no generally practical methods of demonstrating the correctness of complex software systems have emerged. In fact, large software systems generally are *not* correct; nor are their specifications (when they exist). The concept of “software reliability” provides a more applicable notion of software quality. It has been defined as the “probability of failure-free software operation for a specified period of time in a specified environment” [12]. It is useful to generalize this definition to permit different measures of the reliability of a system.

**Definition.** A *reliability measure* for a software system is a quantitative measure of the extent to which the observed functional behavior of the system accords with its desired (but not necessarily specified) behavior.

Examples of reliability measures include: the frequencies of particular types of failures; mean time to failure (MTTF); the mean squared deviation of numerical output from expected output; and the proportion of users experiencing one or more system crashes per month. It is often desirable to characterize the reliability of a multifunctional system using a *suite* of several reliability measures that address different aspects of its behavior.

The analogies between EHR systems, on one hand, and drugs and medical devices, on the other hand, suggests that it is appropriate to characterize the *operational safety* of an EHR system in terms of the frequencies of *adverse events that are actually or potentially harmful to patients*. These measures are defined, directly or indirectly, in terms of a population of system executions or transactions (e.g., treatment encounters) that occurred at one or more deployment sites. Other kinds of evidence, e.g. from fault-tolerant design or formal methods, might be used to argue that a system’s safety is actually greater than is indicated by an upper confidence bound on its adverse event rate. However, neglecting to obtain accurate statistical estimates of the occurrence rates of harmful or potentially harmful events and to consider their implications seriously is not scientifically defensible.

For an EHR system, safety measures should include at least the (separate) frequencies of such distinct events as system crashes and hangs, record displays with erroneous or missing entries, and erroneous or missed treatment orders, alerts, and reminders. System availability and the mean and variance of response time are also clinically important measures. Additional measures, which characterize the safety of specific clinical functionality such as CPOE or CDS are also desirable [13]. Adverse events involving EHR system functionality could be caused by software, hardware, human factors, environmental conditions, or some combination of these factors [14, 15]. Even if investigation reveals a single cause for an adverse event, is appropriate to consider it to be a failure of the overall software-hardware-human system, since patient welfare depends upon the safety and efficacy of the latter.

The *efficacy* of an EHR system can be characterized by measures of the direct and indirect effects of the system on patient outcomes, medication errors, provider costs, efficiency, patient satisfaction, and other variables. For example, a drop in patient mortality that is due to adoption of an EHR system would be one indication of its efficacy. Alternatively, an EHR system’s efficacy can be characterized by measures involving *surrogates* for the outcomes of interest, such as a measure of improved compliance with clinical practice guidelines. Demonstrating that an EHR system is actually responsible for improvements in efficacy measures may be difficult, however. This issue is discussed in Section 4.

## 3. ESTIMATING SAFETY AND EFFICACY MEASURES

Among existing software engineering techniques, *software reliability modeling and estimation* techniques [12] are perhaps most similar to the techniques used to evaluate medical treatments, because the former are intended to characterize the operational reliability of a software system based on statistical analysis of its failure history. Some software reliability models are based on questionable assumptions [16]. One such assumption is that future debugging will change a system’s failure rate according to a probability distribution of known form and

---

<sup>1</sup> Exceptions may be made by FDA for devices that “substantially equivalent” to an already approved predicate device. This practice has been strongly criticized [3].

with estimable parameters. Unfortunately, experience indicates that the actual effects of debugging are so variable as to render any reliability model based on such an assumption unsuitable for use in evaluating safety-critical software.

Fortunately, not all software reliability estimation techniques involve problematic assumptions. It is possible to estimate reliability measures directly from a sample of observations of software behavior in the field [17-19]. For example, the frequency of system crashes can be estimated based on automated crash reports, and the frequency of other failures can be estimated based on reports from (vigilant) users and on recorded execution data. These techniques can be employed to estimate safety measures for EHR systems. For the resulting estimates to be predictive, the sampled behaviors must be representative of future system usage. Thus an assumption of some statistical regularity in behavior of the user population is necessary. Fortunately, this assumption can be checked by collecting statistics that characterize system feature usage over time [20]. It is *not* necessary to actually specify the complex distribution of EHR system inputs.

There are several challenges to estimating EHR system safety measures directly from observed behavior, which can be overcome with appropriate steps. Users may overlook some system failures or neglect to report failures they do observe. Modest incentives may encourage them to report failures accurately. Users can be compelled by law or organizational policy to report all system failures they observe. Even so, it is desirable to instrument a system to record audit logs that permit developers to confirm users' failure reports or to discover failures that users did not report. Ideally, executions would be captured online so as to enable offline replay [21].

Another challenge is the *level of accuracy* needed to adequately demonstrate satisfaction of demanding safety requirements such as, say, a 95% upper confidence bound on the rate of adverse events that is at most 0.001 per patient. Cost is likely to preclude having developers review thousands of captured executions in detail. To obtain sufficiently accurate estimates, it may be necessary to employ data mining techniques, sampling designs, and estimators to exploit *auxiliary information* about executions such as their profiles. This can be done in a way that is *model assisted* [22], rather than model dependent, in the sense that a statistical model is used to improve the efficiency of estimation, yet the estimators used remain (approximately) unbiased and consistent even if the model is flawed. One example is the use of cluster analysis of execution profiles, in conjunction with stratified random sampling, to accurately estimate software failure rates [18]. In this case the "model" is a partitioning of system executions into clusters of similar ones. Other types of models, such as statistical regression models, are also applicable [22].

Finally, the need to reassess safety measures whenever an EHR system is modified engenders additional costs and delays. These can be mitigated by leveraging previously captured inputs and profiles to reduce the costs of re-estimating these measures after changes to the system (or to its operating environment) [23]. An important but somewhat counterintuitive point in this regard is that successfully correcting the defects that caused failures to occur during the process of estimating safety measures *does not* justify reducing the estimated failure rate to zero. Any sound approach to updating an estimate of a safety measure after

maintenance must involve *some additional field testing or testing with captured operational inputs*.<sup>2</sup>

#### 4. EVIDENCE-BASED VALIDATION AND IMPROVEMENT

*Evidence-based medicine* [24] requires evaluating the safety and efficacy of medical treatments and other health interventions by carefully designing and conducting clinical studies and by rigorously analyzing the results using statistical methods. The "gold standard" for evaluating treatments is a *randomized clinical trial (RCT)* [25], in which an appropriate group of volunteer subjects is randomly assigned to receive either a *treatment* under evaluation or a *control* intervention such as a placebo or standard treatment. The patients who receive a given treatment comprise a *treatment group*; the patients who receive the control intervention comprise the *control group*. If multiple treatments are evaluated there are multiple treatment groups. The random assignment of patients to these groups helps to ensure that the groups are relatively *balanced* and therefore *comparable* with regard to possible *confounding variables (confounders)*, which are variables other than the treatment(s), such as age or overall state of health, that may influence patients' outcomes. Random treatment assignment is preferable to systematically balancing treatment and control groups with respect to known confounding variables, because some confounders may be unknown or unobserved.

Although randomized controlled trials are the preferred mechanism for evaluating clinical interventions, they have a number of limitations [25]. For example: it may be unethical to give subjects with a critical illness a placebo; the study volunteers may not be representative of the relevant patient population; the number of suitable volunteers may be too small to achieve desired statistical power; and the conditions of the trial may be far more favorable to a treatment than the conditions under which it would be used in the field. For these reasons, it may be desirable or necessary to conduct an *observational study* [26], in which treatment assignments are *not* deliberately randomized. Finally, an observational study may be based on the existing treatment records of a large sample of patients. EHRs have significant advantages for this purpose [3, 27].

Although some computer-controlled medical devices are evaluated in RCTs, it is probably not feasible to evaluate EHR systems with them. To minimize bias, such a trial would require not only randomly assigning patient records to either an EHR system or a paper record system for storage, but randomly selecting and assigning physicians, or perhaps entire health care organizations, to use each of the two systems. Because patients' outcomes depend on many other factors besides whether they are "treated" with an EHR system, a very large study would be necessary to statistically identify the "treatment effect" of managing a patient's record using an EHR system. It is hard to imagine how such a study could be managed in real health care settings. Moreover, this process surely could not be repeated every time the EHR software was modified, in order to validate the changes.

On the other hand, large-scale observational studies of EHR system safety and efficacy are quite feasible. Such a study would be greatly facilitated by the following steps: (1) equipping the

---

<sup>2</sup> Using captured operational inputs to re-estimate safety measures is sound only if neither the system's interface nor its usage has changed since the inputs were captured.

system with both automatic crash reporting and a convenient mechanism by which users can immediately report other adverse system events and their symptoms; (2) automatically capturing I/O and execution data to be used to confirm adverse events and diagnose their causes; and (3) ensuring proactively that the data recorded in EHRs itself is sufficiently accurate and complete to support research [28], which entails including, when possible, the values of potential confounding variables.

## 5. RESEARCH DIRECTIONS

Steps (1)-(3) above lay the groundwork for research on a number of important aspects of evidence-based validation and improvement (EBVI) of EHR systems.

**Accurate estimation and re-estimation of system safety and efficacy measures.** Prior research on application of “model-assisted” sampling and estimation techniques to software reliability estimation only scratches the surface of what is needed to support EBVI of EHR systems. Further research is needed to determine what sampling designs, what user, EHR, and execution data, and what types of multivariate analysis, statistical models, and estimators are most efficient (with respect to maximizing precision and minimizing sample size and other costs) and practical for use in estimating safety measures for EHR software and in updating estimates after system or environmental changes. A closely related issue is how statistically based techniques for *causal inference* based on observational data [29], can be adapted to permit efficacy measures for EHR systems to be estimated with minimal confounding bias.

**Adverse event detection and confirmation.** Whereas unbiased estimation of safety measures requires a representative sample of system executions, a more aggressive approach is appropriate to root out possible system hazards before they cause serious harm to patients. Such an approach might be based partially on techniques for post-market detection of *adverse drug reactions* [30]. For example, data mining techniques might be applied to execution profiles to correlate unusual but seemingly harmless system behaviors with other events that are known to be hazardous or to correlate unexplained adverse event reports from multiple users.

**Automatic fault localization.** Given that one or more hazardous EHR system failures have been observed, it is imperative to diagnose their cause(s) as rapidly as possible. Statistical and non-statistical *fault localization* techniques (e.g., [31-33]) could expedite this process, especially if their precision can be improved. Possible research directions for achieving such improvement include: employing statistical techniques for causal inference from observational data [29, 34]; mining a combination of clinical data, execution data, and user feedback about failure symptoms to facilitate grouping of related failures and isolation of their causes; and combining evidence from different approaches to automatic fault localization.

**Requirements monitoring and adaptive redesign.** Because of the way EHR system functionalities and workflows are intertwined with clinical practices, the ideas of *requirements monitoring* [35], *usage analysis* [36], and *architecture-based monitoring* [37] are especially relevant to them. Substantial EHR-specific research is needed to articulate an evidence-based approach to system adaptation and improvement, which involves collecting data about both system usage and internal dynamics and analyzing it to inform decisions about system enhancements. An important sub-problem is evaluation of proposed new system features, which *are*, in contrast to entire systems, practical to

evaluate in randomized controlled trials (by installing them in randomly selected system instances).

## 6. REFERENCES

- [1] DesRoches, C. M., Campbell, E. G., Rao, S. R., Donelan, K., Ferris, T. G., Jha, A., Kaushal, R., Levy, D. E., Rosenbaum, S., Shields, A. E. and Blumenthal, D. Electronic Health Records in Ambulatory Care -- A National Survey of Physicians. *N Engl J Med*, 359, 1 (July 3, 2008 2008), 50-60.
- [2] American Recovery and Reinvestment Act of 2009, Stat. Pub. L. No. 111-5, 123 Stat. 115 (2009).
- [3] Hoffman, S. and Podgurski, A. Finding a Cure: The Case for Regulation and Oversight of Electronic Health Record Systems. *Harvard Journal of Law and Technology*, 22, 103 (2008).
- [4] Downing, G., Boyle, S., Brinner, K. and Osheroff, J. Information management to enable personalized medicine: stakeholder roles in building clinical decision support. *BMC Medical Informatics and Decision Making*, 9, 1 (2009), 44.
- [5] Leveson, N. G. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.
- [6] Walker, J. M., Carayon, P., Leveson, N., Paulus, R. A., Tooker, J., Chin, H., Bothe, A. and Stewart, W. F. EHR Safety: The Way Forward to Safe and Effective Systems. *Journal of the American Medical Informatics Association*, 15, 3 (May 2008 2008), 272-277.
- [7] Butler, R. W. and Finelli, G. B. The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software. *IEEE Trans. Softw. Eng.*, 19, 1 (1993), 3-12.
- [8] Horwitz, S., Reps, T. and Binkley, D. Interprocedural slicing using dependence graphs. *ACM Trans. Program. Lang. Syst.*, 12, 1 (1990), 26-60.
- [9] Haynes, A. B., Weiser, T. G., Berry, W. R., Lipsitz, S. R., Breizat, A. H., Dellinger, E. P., Herbosa, T., Joseph, S., Kibatala, P. L., Lapitan, M. C., Merry, A. F., Moorthy, K., Reznick, R. K., Taylor, B. and Gawande, A. A. A surgical safety checklist to reduce morbidity and mortality in a global population. *N Engl J Med*, 360, 5 (Jan 29 2009), 491-499.
- [10] Hoffman, S. and Podgurski, A. In *Sickness, Health, and Cyberspace: Protecting the Security of Electronic Private Health Information*. *Boston College Law Review*, 48, 331 (2007), 331-386.
- [11] Dijkstra, E. W. *A Discipline of Programming*. Prentice Hall PTR, 1997.
- [12] Lyu, M. R. *Handbook of software reliability and system reliability*. McGraw-Hill, Inc., City, 1996.
- [13] Classen, D. C., Avery, A. J. and Bates, D. W. Evaluation and Certification of Computerized Provider Order Entry Systems. *Journal of the American Medical Informatics Association*, 14:(January 1, 2007; 2007), 48-55.
- [14] Ash, J. S., Sittig, D. F., Dykstra, R. H., Guappone, K., Carpenter, J. D. and Seshadri, V. Categorizing the unintended sociotechnical consequences of computerized provider order entry. *International Journal of Medical Informatics*, 76, Supplement 1 (2007), S21-S27.
- [15] Harrison, M. I., Koppel, R. and Bar-Lev, S. Unintended Consequences of Information Technologies in Health Care--

- An Interactive Sociotechnical Analysis. *Journal of the American Medical Informatics Association*, 14, 5 (2008), 542-549.
- [16] Alan, W. *Software Reliability Growth Models: Assumptions vs. Reality*. City, 1997.
- [17] Brown, J. R. and Lipow, M. Testing for software reliability. *SIGPLAN Not.*, 10, 6 (1975), 518-527.
- [18] Podgurski, A., Masri, W., McCleese, Y., Wolff, F. G. and Yang, C. Estimation of software reliability by stratified sampling. *ACM Trans. Softw. Eng. Methodol.*, 8, 3 (1999), 263-283.
- [19] Currit, P. A., Dyer, M. and Mills, H. D. Certifying the reliability of software. *IEEE Trans. Softw. Eng.*, 12, 1 (1986), 3-11.
- [20] Hilbert, D. M. and Redmiles, D. F. An approach to large-scale collection of application usage data over the Internet. In *Proceedings of the Proceedings of the 20th international conference on Software engineering* (Kyoto, Japan, 1998). IEEE Computer Society.
- [21] Steven, J., Chandra, P., Fleck, B. and Podgurski, A. jRapture: A Capture/Replay tool for observation-based testing. *SIGSOFT Softw. Eng. Notes*, 25, 5 (2000), 158-167.
- [22] Sarndal, C.-E., Swensson, B. and Wretman, J. *Model Assisted Survey Sampling*. Springer-Verlag, 1992.
- [23] Podgurski, A. and Weyuker, E. J. Re-estimation of software reliability after maintenance. In *Proceedings of the Proceedings of the 19th international conference on Software engineering* (Boston, Massachusetts, United States, 1997). ACM.
- [24] Evidence Based Medicine Working Group. Evidence-based medicine: A new approach to teaching the practice of medicine. *JAMA*, 268, 17 (Nov. 4 1992), 2420-2425.
- [25] IOM. *Initial National Priorities for Comparative Effectiveness Research*. Institute of Medicine, 2009.
- [26] Rosenbaum, P. R. *Observational Studies*. Springer, 2002.
- [27] Stewart, W. F., Shah, N. R., Selna, M. J., Paulus, R. A. and Walker, J. M. Bridging The Inferential Gap: The Electronic Health Record And Clinical Evidence. *Health Aff*, 26, 2 (March 1, 2007 2007), w181-191.
- [28] Terry, A. L., Chevendra, V., Thind, A., Stewart, M., Marshall, J. N. and Cejic, S. Using your electronic medical record for research: a primer for avoiding pitfalls. *Fam. Pract.*, 27, 1 (February 1, 2010 2010), 121-126.
- [29] Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009.
- [30] Classen, D. C., Pestotnik, S. L., Evans, R. S. and Burke, J. P. Computerized surveillance of adverse drug events in hospital patients. *Quality and Safety in Health Care*, 14, 3 (June 2005 2005), 221-226.
- [31] Jones, J. A., Harrold, M. J. and Stasko, J. Visualization of test information to assist fault localization. In *Proceedings of the Proceedings of the 24th International Conference on Software Engineering* (Orlando, Florida, 2002). ACM.
- [32] Liblit, B., Naik, M., Zheng, A. X., Aiken, A. and Jordan, M. I. Scalable statistical bug isolation. In *Proceedings of the Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation* (Chicago, IL, USA, 2005). ACM.
- [33] Zeller, A. Isolating cause-effect chains from computer programs. *SIGSOFT Softw. Eng. Notes*, 27, 6 (2002), 1-10.
- [34] Baah, G. K., Podgurski, A. and Harrold, M. J. Causal inference for statistical fault localization. In *Proceedings of the Proceedings of the 19th international symposium on Software testing and analysis* (Trento, Italy, 2010). ACM.
- [35] Cohen, D., Feather, M. S., Narayanaswamy, K. and Fickas, S. S. Automatic monitoring of software requirements. In *Proceedings of the Proceedings of the 19th international conference on Software engineering* (Boston, Massachusetts, United States, 1997). ACM.
- [36] Hilbert, D. M. and Redmiles, D. F. Extracting usability information from user interface events. *ACM Comput. Surv.*, 32, 4 (2000), 384-421.
- [37] Cheng, S.-W., Garlan, D., Schmerl, B. R., Sousa, P., Spitznagel, B., Steenkiste, P. and Hu, N. Software Architecture-Based Adaptation for Pervasive Systems. In *Proceedings of the Proceedings of the International Conference on Architecture of Computing Systems: Trends in Network and Pervasive Computing* (2002). Springer-Verlag.