

GLIGOR83 Virgil D. Gligor and Gary L. Luckenbaugh, "An Assessment of the Real-Time Requirements for Programming Environments and Languages", (Proceedings of the Real-Time Systems Symposium, IEEE, December 1983, pp. 3-19). This paper starts with a claim that real-time programming differs greatly from concurrent programming. To substantiate the claim, the authors give a couple of examples, and then provide indications how a programming language would aid in the task of writing real-time programs. Next, the authors state what they believe to be requirements for real-time programming. These are control requirements, timing requirements, distributed systems requirements and "Good programming" requirements. Control requirements basically encompass process synchronization and precedence-constraint description of real-time processes. Timing requirements mean that guaranteed schedulability must be assessed for each real-time process as well as the entire real-time system. Requirements of distributed systems ask that distributed interprocess communication must be efficient and reliable. "Good Programming" language requirements are that the language must be simple, and it must have both procedural and data abstractions, allow for separate compilation, and include testing and debugging tools. Having defined real-time programming requirements, Gligor and Luckenbaugh discuss seven concurrent languages in an attempt to assess their suitability for real-time programming in distributed systems. Three of the Languages, Pearl, Modula-2 and Edison, are then labeled as being 'among the best languages for real-time programming' and picked as winners of this survey. Ironically, none of the languages considered provide any support for distributed programming, and only two, Pearl and Modula-2, have a chance of being suitable for real-time control applications! After the survey, the authors briefly introduce a number of validation techniques for real-time systems and correctly conclude that too much validation work has been done on theoretical and experimental 'toy' models, as opposed to incorporating automatic/manual validation into live real-time systems. In their paper, Gligor and Luckenbaugh have talked a fair bit about various real-time issues and requirements. The concepts discussed are quite interesting, contemporary and challenging. Their real-time language survey, however, leaves much to be desired. Only one language, Pearl, truly intended for real-time programming was included in the survey, yet three winners were selected. Moreover, introducing real-time validation after the survey seems of little value, for ease of program validation was not a criterion on which the languages were compared. All in all, the paper is far from being one of the best ones written on the subject, yet the authors should be commended on bringing the distinctions between real-time programming and concurrent programming to the public.