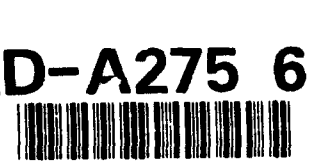
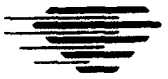


Technical Report
CMU/SEI-93-TR-29
ESC-TR-93-203

(2)



Carnegie-Mellon University
Software Engineering Institute

Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)

Priscilla Fowler
Linda Levine

December 1993

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment or administration of its programs on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state or local laws, or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state or local laws, or executive orders. While the federal government does continue to exclude gays, lesbians and bisexuals from receiving ROTC scholarships or serving in the military, ROTC classes on this campus are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pa. 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pa. 15213, telephone (412) 268-2056.

Technical Report
CMU/SEI-93-TR-29
ESC-TR-93-203
December 1993

Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)

Priscilla Fowler
Linda Levine

Transition Models

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESC/ENS
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.
This report was funded by the U.S. Department of Defense.

Table of Contents

List of Figures

Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)

Abstract: This case study reports on efforts to transform rate monotonic scheduling theory from an academic theory into a practical analytical technique and to transition that technique into routine practice among developers and maintainers of software embedded in real-time systems.

1 Introduction

Rate monotonic analysis (RMA) is a simple, practical, mathematically sound way to guarantee that all timing requirements will be met in real-time systems. RMA allows engineers to understand and predict the timing behavior of real-time software to a degree not previously possible. The Rate Monotonic Analysis for Real-Time Systems (RMARTS) Project at the Software Engineering Institute (SEI) has demonstrated how to design, implement, troubleshoot, and maintain real-time systems using RMA. From 1987-1992, the project worked to develop the technology and encourage its widespread use to reduce risk in building real-time systems.

The acquisition and introduction of new software technologies (including tools, methods, and management approaches) is so much a part of most software development and maintenance efforts that we do not even call it out as a separate activity. However, one reasonable explanation for why cost and schedule overruns are so common in software projects is the continual learning required on the part of software engineers and managers. One solution to the software "crisis" is to better understand and anticipate problems and barriers in the introduction of new software technologies. (This is the focus of the Transition Models Project.[1])

This case study describes efforts of the RMARTS Project to transform rate monotonic scheduling theory from an academic theory into rate monotonic analysis—a practical technique for analyzing existing systems and designing new ones. Of particular importance is how the RMARTS Project transitioned this technology to the community of potential users. This document is Part 1 of the study. It examines how problems of introduction, learning, and use were anticipated and addressed during the development of the technology. Part 2 (CMU/SEI-93-TR-30) describes the experiences of several projects in one organization in attempting to adopt and apply RMA.

The investigation of RMA transition activities is intended to make a twofold contribution to greater understanding of technology transition.[2] Our aim is to encourage researchers to

---

[1] The study of the maturation of RMA is one part of a larger effort to build a "thick description" [Geertz 1972] of software technology transition. (Such an ethnographic description makes the tacit explicit; attends to cultural practice, including communication in a given community, organization, or group; attends to detail and not only abstract concepts; and captures language as it is in use—as it reveals the values and priorities of those groups.) Additional work in the Transition Models Project explores related levels of analysis; for example, Fowler & Levine [1993] offer a conceptual framework for technology transition, from the birth of a technology to its retirement.

further explore the precepts presented here with respect to other software technologies and to help practitioners learn from and apply strategies used by the RMARTS team. Practically speaking, people working in the area of technology transition should be able to adapt the heuristics identified here with respect to another software technology. The case study approach is a good match for our double purpose: the research method allows for close examination and interpretation; and, the detailed case study form can provide practitioners with surrogate experience of a complex transition situation.

As indicated, the case study consists of two technical reports: Part 1 (this document) is concerned with the analysis of the transition activities according to phases of a technology maturation life cycle; and Part 2 investigates the processes of adoption and implementation. Together, the two parts allow us to attend to development and user perspectives—or more colloquially put, to technology *push* and technology *pull*. Part 1 of the case study includes these sections: a brief description of RMA; the rationale for selecting it as a topic of study; descriptions of several technology maturation models; research method and procedure; results; and implications and directions for future research.

The data used for Part 1 of the study are largely drawn from the reports of RMARTS Project accomplishments in the annual SEI one and five year plans. Transition activities from 1987 through the first half of 1993 were analyzed and coded according to phases of a technology maturation life cycle. The coded data were then corroborated by supporting artifacts and by RMARTS Project members' reviews of the analysis. Additional information was collected through financial records, attendance at project meetings, and interviews. A more detailed description follows in the *Method* section of this report (Section 5, page 15).

---

[2] The phrase "technology transfer" is usually preferred, except within the DoD. For the purposes of this report, we consider "technology transfer," "technology transition," and "technology deployment" to be synonymous. In addition, we agree with Tornatzky & Fleischer [1990]: "Technology transfer, while a commonly used term, has a host of nuances, not the least of which is the image that technology is something that is physical, comes in large crates or on pallets, and gets literally moved from place to place." On this basis, they "use the more inclusive and less encumbered notion of *deployment* (p. 118; italics Tornatzky & Fleischer)"; we prefer "technology transition."

2 Rate Monotonic Analysis

RMA helps software engineers who are designing, building, troubleshooting, and maintaining real-time systems to understand and predict the timing behavior of hard real-time systems to a degree not previously possible. Real-time systems are often seen as a "niche" within the software world, but they are a critical niche. Real-time software is often embedded in life-critical systems such as avionics and other transportation systems, patient monitoring equipment, and process control systems in chemical processing and nuclear power plants.

In the software embedded in such real-time systems, multiple tasks contend for the use of a finite amount of resource—for example, of the central processing unit (CPU). Typically these tasks, such as monitoring altitude, monitoring cabin pressure, or controlling fuel injection level on an aircraft, are of differing priorities and require different amounts of CPU effort to complete their work. These tasks can occur both at regular intervals and irregularly. Real-time systems must complete critical tasks (for example, the lowering of the landing gear) by particular deadlines, or place the entire system at risk. Without the appropriate handling of schedules and priorities, a lower priority task of relatively long duration—for example, intermittent monitoring of passenger cabin pressure— can monopolize the CPU at the expense of a critical task.

Traditionally, the approach to calculating appropriate task mixes has been by trial and error. Programs are written to accomplish the required tasks, but until the tasks are integrated into a system and tested as a system, there is no way to know whether all tasks can be accomplished within the constraints of the available CPU. Most often deadlines are not met until after many iterations of testing, program revision, and more testing. For these reasons, real-time systems have earned the reputation of being expensive, behind schedule, risky, and difficult to maintain.

While the traditional approach is manageable for simple systems, especially when particular system designers have become expert in the successful design of task mixes, it is unmanageable for large systems. As large real-time systems are increasingly built from commercial "off-the-shelf" or separately contracted subsystems, handcrafting of scheduling is increasingly risky. Rate monotonic scheduling theory and its method of application, RMA, provide a scientific approach that can be used, before system integration, to determine whether schedule requirements can be met and how well, and under what conditions task completion can be guaranteed.[3] Because RMA solves a difficult problem early in the development of a real-time system, it is an important innovation with respect to technical work and its management—it can result in significant savings not only of CPU but potentially of both system development resources and operational resources such as hardware.

---

[3] The early focus of RMA ensured that as long as CPU resource utilization lies below a certain bound (generally a percentage of total possible resource), all the timing requirements will be met [Sha & Goodenough 1990]. More recently, the focus of RMA has shifted to computing the worst-case response time and comparing it to the *response time requirement*.

3 Rationale and Background

The transformation of rate monotonic scheduling theory into RMA and its reduction to practice did not happen by accident. The original Carnegie Mellon University (CMU) team (the Advanced Real-Time Technology Project) and the subsequent alliance of SEI and CMU teams have focused on the key problems that emerged from the development of national projects. In fact, the selection of the theoretical foundation for real-time system development as an area of study originated with the need to solve problems with the development of the Navy BSY-1 system.[4]

The Software Engineering Institute, a federally funded research and development center (FFRDC) at CMU, is well positioned to explore interactions between theory and practice. Its engineering focus seeks a balance between ease of implementation (typically the concern of practice) and analyzability (a key theoretical issue). Such interaction between theory and practice is necessarily iterative. When theory is focused on actual problems encountered in development, the problem is well set and well defined; work can then be done, iteratively, to test and refine the theory.

Within the SEI and its constituencies, the development of RMA and the work of the RMARTS Project is considered an exemplary model of the interaction between theory and practice. Satisfaction of three additional criteria also influenced our decision to study RMA:

1. The characteristics of the technology itself—including its size, observability, and maturity.
2. The qualities and qualifications of the change agents involved.
3. The accessibility of the history of the technology.

We discuss each of these in turn.

3.1 Characteristics of the Technology: Size, Observability, and Maturity

Studying the transition of software technologies is difficult for a number of reasons. First, the process-intensive nature of software technologies means complex interaction between the social context and the technical content of the technology [Tornatzky & Fleischer 1990].The challenge of studying transition in an organizational setting is well documented [Nord & Tucker 1987, Yin 1984]. Second, the adoption of many software technologies occurs over an extended time period, at least months and often years. Third, technology introduction has an impact on not only the technical subsystem, but the managerial, strategic, human, and cultural subsystems as well [Morgan 1986].

---

[4] The paper, "Rate Monotonic Analysis for Real-Time Systems" by Sha, Klein & Goodenough [1991], reviewed some important historical decisions in the development of RMA.

RMA requirements for effective application are relatively limited. For example, while RMA does require engineers to reframe their understanding of scheduling issues to a more abstract level, only moderate training is required for people to be effective in using the technology.[5] It can be adopted by an individual engineer as part of his or her approach to designing or analyzing systems; it can also be applied at almost any point in time in system development or maintenance. RMARTS Project members recount how they are able to quickly demonstrate, in consulting and classroom settings, the utility of the approach.

According to Adler & Shenhar [1990], adopting a technology that will change skills and procedures can be accomplished typically within the space of weeks; in contrast, adopting a technology involving a change in either structure or strategy requires months of planning and implementation. RMA can be incorporated into software engineering processes with relative ease over a period of several months. In this regard, we classify the technology as "small."[6] The initial stages of user commitment to the use of RMA—contact, awareness, understanding, and trial use [Conner & Patterson 1982]—can be observed within a short time. To use RMA, neither managers nor engineers must adjust their paradigm of software development significantly.

These same factors make RMARTS transition readily "observable" [Rogers 1983] within a reasonable period of time. RMA can be adopted incrementally: its adoption can range from application to an existing system by one engineer to application across an entire division as standard practice in designing new systems. A project team or group of project engineers can adopt and implement RMA within a few months. This means transition can be studied not only retrospectively but "in process,"[7] and that process is relatively short-term. In addition, because RMA is a technology without extensive "cultural" content (in contrast with a technology such as Total Quality Management), its transition process would be less muddied by major shifts in attitude and belief systems.[8]

Finally, the maturity of RMA is important. Given the process-intensive nature of software technologies [Tornatzky & Fleischer 1990], less mature technologies are likely to have poorly de-

---

5. One project member and former resident affiliate stated in an interview that over a period of time, his use of RMA had caused a shift in his view of architectures. He began to see an important distinction between "architectures" and "attributes of architectures." He noted that he began to look at "software performance in terms of preemption, computation, and blocking." (These are concepts used in RMA theory.)

6. Compare this type of technology with, for example, a CASE tool that may require adjustment to interfaces with other software, upgraded or new hardware, and revised documentation standards. In this respect, we classify RMA as "small." For a preliminary taxonomy of software technology types, see Fowler & Levine [1992a]. We emphasize that the type of a technology (and its related size) represents only one dimension of a technology's profile. The maturity of the technology and its ease of implementation—in other words, the degree to which it is a "whole product" [Moore 1991] are also critical dimensions to the technology's complete profile.

7. Studying the in-process adoption of RMA is not the subject of this report; however, future investigations may lead in this direction.

8. Any technology has some cultural content, in the sense of requiring an adjustment in the user's belief systems and behavior. For example, RMA requires acceptance of logical concurrency, a shift for most software engineers and their managers, with some implications, albeit limited, for the structure and scheduling of their work. It does not, apparently, require restructuring of the software project management process or of reward systems. See Part 2 of this case study for more information.

fined transition processes. The limited impact of RMA on the structure and work processes of the organization makes it somewhat easier to separate technical problems from problems in the transition approach. In fact, while RMA is not yet fully mature in the sense of a commercial *whole product* [Moore 1991] that incorporates the secondary products and services that late majority adopters need (such as training and support, courses, documentation, handbooks, etc.), as an analytical method RMA is no longer evolving rapidly.[9]

3.2 Change Agent Qualities and Qualifications

Studying less as well as more successful cases is productive; however, we begin with a technology transition effort that was considered a success. This was so with RMA within the SEI and its larger context; and part of its success has been attributed to the qualifications of the change agents associated with the technology. The RMARTS Project was well endowed with senior personnel who, while not trained as change agents, were highly credible with their peers and well-schooled in the politics of professional and industry associations. Collectively, their capabilities beyond software engineering and computer science included project management, consulting, acquisition, and research. We elaborate on some of these capabilities to illustrate how change agent qualifications impacted transition.

In terms of basics, several members of the project were exposed to ideas about transition early in the life of RMARTS. This exposure included: awareness of case studies of software technology transition [Redwine et al 1984]; an understanding of the nature of resistance to technological change from material in the Managing Technological Change Workshop, taught at the SEI since 1988; and discussion of technology advocates and receptor functions[10].

The RMARTS staffing turnover was minimal during the five years of the project's official existence (1988 - 1992),[11] since turnover was mainly limited to resident affiliates (personnel on temporary assignment from industry or government organizations). John Goodenough and Lui Sha, cofounders of RMARTS, both had a strong interest in the subject of technology transition. Their plan for transitioning RMA technology was not detailed or formal, but consisted of innumerable course corrections toward a vision of the world post-institutionalization of RMA. In 1991, roughly halfway through the life of the project, RMARTS developed a "stage model of technology transition" to explain and describe what the project had done and was planning to do.[12] In the following section, *Technology Life Cycle Models*, we describe this model in greater detail. We also discuss other models that are consistent with the research and development (R & D) life cycle.

Goodenough was cognizant of application domains within the world of real-time systems that would have use for RMA. Sha recruited resident affiliates from key domains. His purpose was

---

9. There are efforts underway to encapsulate and guide the use of the RMA method with software tools.

10. These concepts were documented [Fowler 1990], and were commonly presented to new SEI resident affiliates as part of their orientation as early as the winter of 1988. (Personal communication, Tom Ralya, October 1991.)

11. Activities took place before 1988 and after 1992; however, the project was an approved SEI project during the dates cited.

twofold: to provide persons who could carry the technology back to their home organization, and who would regularly provide a skeptical, real-world view of limitations of the technology that might prevent its application and acceptance. In particular, the affiliates helped identify technical barriers to adoption, thereby keeping RMARTS attuned to transition issues.[13] For example, one resident affiliate was not convinced that the Ada programming language could work for real-time systems, and he often played devil's advocate.

A critical aspect of the transition strategy was convincing industry standards committees to support, or at least not preclude, the use of RMA technology in applications. Staff selected for the RMARTS Project included several individuals with experience in real-time systems who had been active in, or had credibility with, industry standards groups. These individuals became members of the standards groups, and influenced the direction of key standards, such as the IEEE FutureBus+. Without the standards work, RMA could have been adopted and applied by individual engineers; with standards, broad use of RMA within real-time systems is becoming a real possibility.

3.3 History

Extensive and largely nonproprietary documentation on RMA was available, both formal and informal, from which case study material could be drawn.[14] We have already mentioned the availability of project members whose tenure with the project was long. In addition, Carnegie Mellon University faculty who evolved the original [Liu & Layland 1973] rate monotonic scheduling theory from which RMA was derived remain at the university. Former resident affiliates and personnel from organizations that cooperated with RMARTS to test RMA were also available for interview.

In sum, the selection of RMA as a transition case study was based on the characteristics of the technology—its size, observability, and maturity—the qualities and qualifications of the change agents involved, and the availability of individuals and documents that helped us trace the development of the emerging technology.

[12] The RMARTS' model consists of five stages: promising technology selected, key limitations addressed, value and transitionability demonstrated, self-sustaining transition, and widespread use. More recently, Goodenough has expressed a preference for changing the name of stage two (originally, key limitations addressed) to: barriers to adoption removed. While "key limitations" often constitute barriers to adoption, he notes "this deserves more emphasis early on: 'key limitations addressed' sounds like academic polishing rather than what we (RMARTS) attempted to do." (Personal correspondence with John Goodenough, October 1993.) For the purpose of this case study, we use the RMARTS model; however, we note that their representation resembles that of others [Botkin & Matthews 1992, Redwine et al 1984, Tornatzky & Fleischer 1990].

[13] Resident affiliates were also recruited from universities to begin to develop a cadre of researchers who would extend RMA and related theory.

[14] Formal deliverables include: reports required of the SEI by contract, technical reports, and external publications; informal documentation includes electronic mail, meeting minutes, and presentation material.

4 Technology Life Cycle Models

Models of R & D represent only one part of the life of a technology. In this section, we discuss the technology life cycle (Figure 4-1); a more specific R & D life cycle; and, finally, we describe the RMARTS Project stage model—a model that synthesizes issues of technology development with issues of technology transition. In other words, we begin with the larger context of technology transition, narrow the focus to R & D activity, and then consider the RMARTS Project's specific model. The latter is our primary concern.

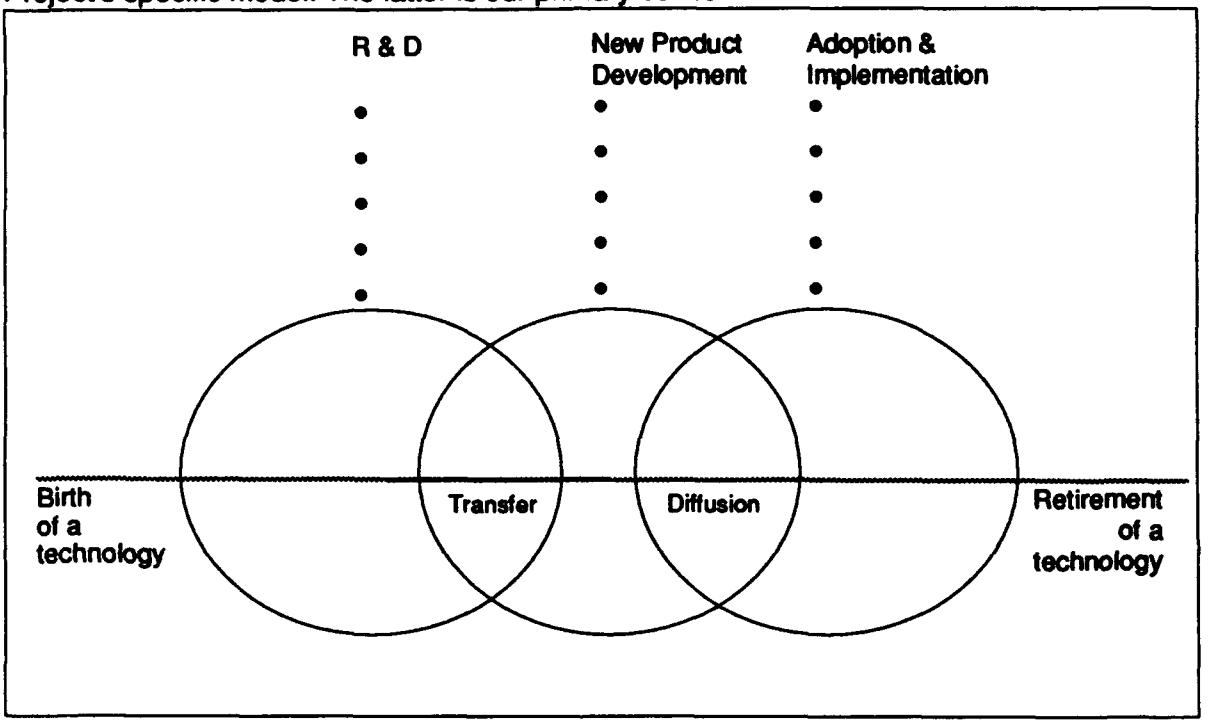


Figure 4-1: Technology Life Cycle

In the course of investigating technology transition as understood by disciplines such as management science, political science, communication, and economics, we have discovered three major perspectives or life cycles. These are: R & D (including the creation of prototypes), new product development, and technology adoption and implementation. Technology transition occurs throughout technology development from the birth of a technology until its retirement. For example, technology that has been commercially developed and is in use in an organization has most likely been transitioned at least twice, between communities respectively concerned with R & D, new product development, and adoption and implementation. In addition, the technology is transitioned as it progresses through its life cycle within each community [Fowler & Levine 1993]. Traditionally, these communities have only limited interaction with each other.

R & D, then, represents only one part of the larger technology life cycle: the focus of R & D is predominantly on the changes that the technology itself goes through as it matures [Botkin &

Matthews 1992, Moore 1991, Tornatzky & Fleischer 1990]. Redwine's representation [1984] of the R & D life cycle is typical.[15] The life cycle includes

- Concept formulation
- Development and extension
- Enhancement and exploration (internal)
- Enhancement and exploration (external)
- (Early) popularization.[16]

The emphasis here is primarily on technology "push," and the perspective is that of the researcher or developer. From this perspective, transition means orchestrating the development of the technology by "moving" it systematically through stages of development until it is finally incorporated into a prototype product.[17]

During the development of RMA, the RMARTS Project looked to R & D models for guidelines or heuristics for accomplishing successful transition. During interviews with RMARTS Project members (in 1992), they reflected on their earlier considerations (through 1990-1991) of the life cycle described above [Redwine et al 1984]. RMARTS staff had seen the model to be generally relevant but of limited use for their purposes. Further, they noted that some phases seemed to blur—for example, the two phases concerning enhancement and exploration. Other phases, they remarked, were difficult to operationalize—how did you actually achieve popularization? In 1990-91, RMARTS was trying to find a way to characterize its own activities. These activities could then be understood at the Software Engineering Institute, and potentially they could provide a model for others to follow. The RMARTS Project preference was for an active model, one that would identify classes of activity to be performed, and related aspects of motivation. Such a model would help throughout development and transition to answer the question: what do you do in order to push and in order to facilitate pull?

---

15. The following is another example. In working on an early version of this study, during Fall 1991, we exchanged informal correspondence with Neil Eastman (of IBM), a member of the SEI Board of Visitors. At that time, Eastman referred to five stages: research activity, technology selection, technology engineering, technology employment, and practice.

16. Conventionally, the R & D life cycle includes "early popularization" [Redwine et al 1984]. However, given what we now know about technology transition and the three interlocking life cycles, we would argue that early popularization occurs as a part of new product development.

17. A good example of this is the U.S Department of Defense funding process with different types of funding for different stages of technology maturation. Research activity moves through the following: basic (university) research, exploratory research, applied research, applied research associated with a specific program, developmental research, etc.

Continued discussion led the RMARTS Project to develop its own model in 1991. The RMARTS transition model consisted of five stages:

- Stage 1: Promising technology selected
- Stage 2: Key limitations addressed
- Stage 3: Value and transitionability demonstrated
- Stage 4: Self-sustaining transition
- Stage 5: Widespread use in target population

For the SEI, an FFRDC, constrained by law to 250 members of the technical staff, the issue of achieving widespread use or popularization (of multiple technologies, not only RMA) represents a significant challenge. The Institute's mission, to advance the state of the software engineering practice, requires a special set of technology transition strategies. The most controversial ideas to surface, then, with respect to the RMARTS model, concerned "transitionability" and the necessary mechanisms to harness natural phenomena (as depicted in conventional R & D models). RMARTS Project members stated that they felt they achieved a breakthrough with the notion of "leveraging the infrastructure" and the related idea of "self-sustaining transition." Essentially, self-sustaining transition involved developing a community of partners who would help create and service the market for RMA.

Briefly summarized, the RMARTS transition model for RMA is as follows[18]: *Stage 1, promising technology selected*, involves working with a customer to find a technology that solves a real problem. In 1982, IBM came to CMU with a problem that led to the exploitation of rate monotonic scheduling theory. During *Stage 2, key limitations addressed*, RMARTS worked with users to ensure that real problems were being solved and worked with theoreticians to find technical solutions. Identifying and removing barriers was critical here. In addition, the project developed tutorial materials as trial transition vehicles. RMARTS worked on their own and with IBM at this stage starting in late 1987. *Stage 3, demonstrate value and transitionability*, involves demonstrating value by solving or discovering problems. The project developed success stories (working with IBM and NASA) and did pilot transition to develop and demonstrate the successful approach.[19] These efforts started in late 1989. In *Stage 4, self-sustaining transition*, RMARTS began to develop the market for RMA. This is the stage that RMA is presently at.[20] Self-sustaining transition for RMA means that others are teaching RMA; others are providing consulting services; compiler vendors support rate monotonic algorithms; hardware support encourages use of RMA; procurement organizations see RMA as a plus; and national

---

18. The summary description that follows draws heavily from a presentation on Rate Monotonic Analysis for Real-Time Systems by John Goodenough. The presentation was an information seminar at the Software Engineering Institute, March 1991.

19. A premise at work here was: a solution to a problem is useless if you can't get people to adopt it. The project focused on refining adoption strategies and developing transition support materials.

20. While the RMARTS Project officially closed down at the end of 1992, Ray Obenza, member of the Products and Services Planning function, has taken the lead on efforts relating to self-sustaining transition. The first RMA Users' Forum was held in September 1992; the second forum was held in November 1993. In the summer of 1993, Kluwer Academic Publishers released *A Practitioner's Handbook for Real-Time Analysis* by Klein, Ralya, Pollak, Obenza, and González Harbour. Several tool builders are planning to develop tools for RMA.

standards allow and encourage the use of RMA. *Stage 5, widespread use*, where the technology is routinely used by a significant portion of the targeted end-user population, remains ahead. For RMA, the targeted group includes: real-time system designers and developers, developers of runtime systems, teachers and developers of academic courses on real-time theory, those performing academic research leading to improved theory; and those performing engineering improvements outside of the SEI. These targets (or measures of success) were identified during the planning stages of the project, not *post hoc*.

Three critical issues must be noted with respect to the RMARTS transition model. First, the model compresses the larger technology life cycle model based on the three interlocking life cycles of R & D, new product development, and adoption and implementation. Second, the model accomplishes this compression by relying on and leveraging an existing infrastructure. For example, instead of building a product itself, the SEI secures a partnership with a new product developer, such as a vendor or tool builder, and hands off technical materials to enable product development.[21] Finally, the RMARTS Project's awareness of the need to operationalize and extend a conventional R & D model reveals their tacit understanding of what is described in business terms as the "value chain."

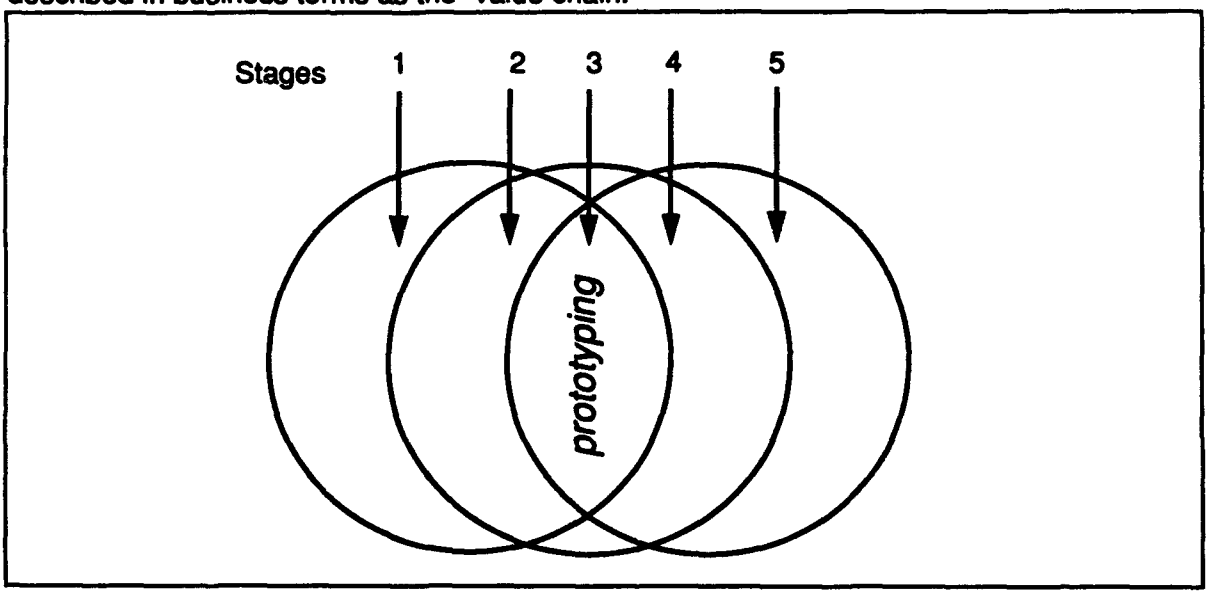


Figure 4-2: Concurrent Transition with Stages

---

[21] Compression of the larger technology life cycle can be described as concurrent technology transition, an approach whereby R & D, product development, and implementation issues are explored simultaneously by means of prototyping, testing, and gathering feedback through alliances between developers and end users. Concurrent technology transition has the potential to significantly reduce the time it typically takes to bring a radical technology (possibly also incremental advancements) to market. The means by which the SEI compresses the technology life cycle and leverages the infrastructure is tied to the Institute's identity as an FFRDC. The principle of leverage is also applicable for "for profits," most likely through strategic alliances. For more on vertical alliances, see "Technology Fusion and the New R & D" by F. Kodama, 1992, *Harvard Business Review*, July-Aug., pp. 70-78.

According to Botkin & Matthews [1992], the value chain describes the process by which a new idea gets to market. "The value chain is a sequence of activities during which value is added to a new product or service as it makes its way from invention to final distribution. When a commercially valuable idea takes forever to get from concept to marketplace—or never arrives—the problem is often a weak or missing link (p. 26)." The value chain is composed of several linked stages, which can then be grouped into three phases:

- Phase 1: research, development, design
- Phase 2: production (manufacturing, fabrication)
- Phase 3: marketing, sales, distribution

One key way to navigate the value chain is through partnerships.[22] Botkin & Matthews [1992] illustrate how large businesses may be weak innovators and/or slow in getting products to market; nonetheless, these bigger corporations can offer smaller partners "stability and credibility, established marketing and distribution channels, and financial resources that are almost unimaginable to strapped young companies (p. 32)." Ideally, companies specializing in one phase of the value chain would partner with other companies able to complete another phase of the process. While RMARTS did not map a specific path to commercialization, the project intuitively endorsed principles of the value chain and partnering through their notion of self-sustaining transition. Self-sustaining transition, we recall, involved developing a viable market for RMA so that others would provide consulting services, ensure that compiler vendors supported RMA algorithms, build tools, etc.

[22] Botkin & Matthews [1992] discuss three additional factors that influence innovation along the value chain: the drive for technological breakthroughs, total quality management (TQM) efforts, and customer focus programs.

5 Method

5.1 Design

In empirical research, it is conventional to talk about triangulation and to design studies that use converging methods and measures to investigate a question or hypothesis. Triangulating is one way to insure reliable data collection and reduce threats of invalidity. The concept is particularly suggestive for research on technology transition. A complete understanding of technology transition requires that one pay attention to multiple units of analysis, including the technical, institutional, and cultural grounds for the adoption and implementation of innovations. For this reason, the case study of the transition of RMA was conceived of in two parts. As we have explained, Part 1, this report, is concerned with the analysis of transition activities according to phases of a technology maturation life cycle. Part 2 examines the processes of adoption and implementation. Together, the two parts offer a robust picture of the transition of RMA, allowing us to attend to development and user perspectives: to technology "push" and technology "pull." The more technical explanation of how this (two-part) investigation of RMA can be seen in the larger context of case study design follows.

The full study can be described as an embedded single case design [Yin 1984]. In layman's terms, the case study is about the transition of RMA; and the study involves several levels of analysis. The main unit was the technology transition effort as a whole; the two smaller subunits were (Part 1) RMARTS Project transition activities and (Part 2) a subset of adopters' and users' perspectives on the technology. For Part 1 of the investigation, a combination of data collection techniques was used, ranging from analysis of historical documents (including plans, reports, and statements) to informal data collection through the researchers' attendance at RMARTS meetings (from September 1991 to December 1992) and personal interviews conducted with RMARTS staff. The following subsection discusses the materials and procedure, including the coding scheme, used for Part 1 of the case study. A separate report on adopters' and users' perspectives of RMA is contained in CMU/SEI-93-TR-30.

This case study on the transition of RMA is best described as enlightening with respect to our ability to observe and analyze a situation not previously accessible to scientific investigation. In this context, "the case is therefore worth conducting because the descriptive information alone will be revelatory [Yin 1984, p. 43]." To date, the transition of RMA technology has not been the subject of research. Nor have software developer perspectives on transition activities and user perspectives on adoption and implementation been juxtaposed in one case study. In this regard, we break new ground.

As is common with single (embedded and holistic) case studies, we cannot extend our observations about RMA and the RMARTS team to other technologies and other technical projects. Emerging theory about software technology transition must be tested through replication of the findings in a second or third technology case, where the theory has hypothesized that the same results should occur. Such replication logic (attending to both technology and organizational context) plays a key role in case study research and in experimentation, allowing one to

eventually establish "external validity": the ability to generalize beyond the specific instance [Yin 1984; Chadwick, Bahr & Albrecht 1984].

The present investigation should not be confused with a historical study of RMA transition activities and the persons involved with the same. As indicated, the issue explored (in Part 1) is technology push; thus, we concentrate on transition according to the perspectives of the developers and development. In addition, because this perspective is revealed through project documents and events, the observations are those of the project as a "whole" rather than individual members. The study does not capture personal interpretations or records of the RMARTS staff, nor the views of others who are external to the SEI. A history of the technology and/or a social network study attending to influence within and between adopting (and co-developing) communities would offer other dimensions of the RMA story. Such investigations represent promising directions for further research.

5.2 Materials and Procedure

As indicated, the main unit of analysis was the technology transition effort as a whole; and the two subunits consisted of (Part 1) RMARTS Project transition activities and (Part 2) a subset of adopters' and users' perspectives on the technology. At each level of the analysis, multiple data collection techniques were used, ranging from the analysis of historical documents or artifacts (including plans, reports, and statements) to informal data gathering through the researchers' attendance at RMARTS meetings (from September 1991- December 1992) and personal interviews conducted with RMARTS staff.

Primarily, data were collected through the analysis of historical documents. These documents consisted of

- Reports of project accomplishments in the annual SEI one and five year plans, spanning from 1988 (when the Real-Time Scheduling in Ada Project was first reported on for 1987) until the present plan for 1993 (reporting on 1992 accomplishments).
- Two quarterly reports on the first half of 1993.

Transition activities for RMA were coded according to phases of the technology maturation life cycle as represented in the RMARTS transition model. Figure 5-1 shows a sample year of coding, with 1988 data presented. (The complete year-by-year analysis is provided in Appendix A.)

The coding procedure included the following:

- Review of the Real-Time Scheduling in Ada (RTSIA) and later the RMARTS sections of the annual SEI one and five year plan, in which the previous year's accomplishments were listed.
- Separation of the prose descriptions of all consequential[23] accomplishments into discrete activities.
- Assignment of each discrete activity to a life-cycle phase.
- Distribution of copies of the summary of activity assignments to all project members.
- Meetings to review assignments with a minimum of three project members.
- Revision of assignments of activities based on reviews.

The reliability of the coding was tested in two ways. First, where possible, we examined supporting documents, including quarterly and technical reports, project summaries, and journal articles to cross-check accuracy of activities and events; second, RMARTS Project members reviewed our coding efforts for 1987-1993 and the draft case study report. The latter review by RMARTS Project members (who might typically be referred to as "key informants") is important in testing the correct operational measures for the concepts being studied here. Technically speaking, this is referred to as *construct validity*.

Two additional data collection methods must be noted. First, we attended RMARTS Project staff meetings from September 1991 until the close of the project in December 1992. These meetings provided the opportunity to gather informal background on RMARTS activities. In January 1993, the main responsibility for RMA transition was officially handed off to another function within the Institute, to Products and Services Planning (P & SP). From January 1993 until the present, we continued to meet with Ray Obenza, the P & SP staff member dedicated to RMA. Second, in the winter of 1993, we conducted a set of interviews with the RMARTS Project to investigate the evolution of the RMARTS transition model. We were particularly interested in the model's relation to other technology maturation life cycles. These interviews also allowed us to explore project history and strategy and to translate technical material into layman's terms.

---

[23] Each coded activity was assumed to require a minimum of one staff month of effort to accomplish. Most activities required substantially more effort.

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Reports for compiler vendors about implementing algorithms for Ada runtime systems. Development of theory implementation validation and performance tests. Analysis of Real-Time Embedded Systems Testbed (REST) Project's Inertial Navigation System for refinement of theory.	IBM resident affiliate arrives. Naval Weapons Center (NWC) resident affiliate arrives. IBM resident affiliate in collaboration with Sr. IBM engineers prepares material for IBM-internal RMS course. Generic avionics case study developed by IBM and NWC. Naval Weapons Ctr resident affiliate developing generic missile case study. Representation of the avionics system implemented using modified Verdix Ada compiler to determine if predicted performance can be obtained.	Work with hardware bus standards groups (IEEE FutureBus+) to allow effective use of prioritized scheduling algorithms. Verdix implements RMS-required scheduling algorithms in its Ada runtime system. Other compiler vendors experimenting with the algorithms. Work with Ada language maintenance and revision efforts. Documents for project managers. IBM experiment with RTCN at IBM.	One-day tutorial prepared to give to engineers at SEI Affiliates Symposium.

Figure 5-1: 1988 Activities

6 Results

The heterogeneity of RMARTS transition strategies and activities—involving different communities, forums, and mechanisms—makes it necessary to describe RMARTS from a number of perspectives. Thus, the following discussion is divided into key topic areas:

- Vision
- Standards work
- Range and pattern of transition activity

6.1 Vision

The consistency with which RMA progressed through the stages of transition was not serendipitous. Goodenough's and Sha's vision for RMA is stated in the introduction to the RMARTS annual plan. In 1988, the plan reads:

> The overall objective of this project is to transition advanced real-time scheduling theory into routine software engineering practice in the context of Ada. After this transition is accomplished, real-time systems will be developed using a set of theoretically sound real-time scheduling algorithms implemented in a runtime scheduler. These algorithms will guarantee that all timing constraints will be met by the system as long as the use of computing resources is less than a certain bound. In addition, when a transient system overload occurs, deadlines will be missed in a predefined order of mission criticality.

The specificity of the discussion is noteworthy: the vision is operationalized in concrete terms, reflecting what the theory was able to do.

Later in the same plan, we see the following:

> Besides improving the state of the practice of real-time software engineering, this project is intended to demonstrate that the SEI can improve software engineering by reducing the time needed to transition theoretical notions into practice.

This previous remark then continues to appear in subsequent plans.

We are able to observe how the statement of project purpose is tuned, in concert with the evolving vision. In 1989 and 1990, it reads:

> The purpose of the Real-Time Scheduling in Ada Project is to transition a new, analytic approach for designing real-time systems into

> routine software engineering practice, particularly for systems implemented in Ada.

The emphasis on Ada was found to be deflecting the attention of potential users from the fact that the theory did not depend on Ada. Therefore, in 1991 the project name changes from *Real-Time Scheduling in Ada* to *Rate Monotonic Analysis for Real-Time Systems*. Moreover, linking the work to Ada was not helping to make the technology more transitionable; it tended to raise questions that ultimately proved irrelevant to the adoption process. The name change also reflected the project's understanding that the theory had a broader range of application than was originally understood. In particular, the project came to realize that the theory could be used to analyze the behavior of systems that had not been designed or implemented with the theory in mind. In this sense, the project was lucky to have picked a theory that turned out to apply so broadly. Now, in 1991, the purpose statement for the project reads

> ...to improve the state of the practice for real-time systems engineering by providing a solid scientific foundation for dealing with timing behavior of real-time systems.

Rate monotonic scheduling theory is presented as the technology of choice for accomplishing this; and the project is no longer "focused primarily on scheduling algorithms..." because of the realization that the theory could be applied more widely than was previously understood.

The existence of a vision, its operational definition and maintenance, and its concentration on transition were key. The vision (and the activities that inform it) is important for three reasons: its breadth, contribution to project management, and evolutionary nature. First, the vision is tuned to represent the technology in an attractive and broadly applicable light. According to one project member, early interaction at RMARTS staff meetings resulted in an increasingly refined understanding of the audiences for RMA. For example, the team determined that the connection to Ada was important but not primary. The purpose of the project's name change was to broaden the audience and to ensure that application of the technology was not limited to the Ada context.

Second, the consistency and specificity of the vision, even as it evolved, was integral to project management. Based on our observations at RMARTS staff meetings, each member appeared to have internalized the vision and to have made decisions consistent with it. Actions were reported at staff meetings where consistency with the vision was checked informally and feedback was provided. Over time, the effect was that of a coordinated team with individuals working autonomously, all moving toward the common goal. Eventually, the RMARTS stage model of transition was developed and formalized to describe what the project had done and was planning to do.

Finally, the vision was not static or cast solely in terms of the state of the technology; the vision grew along with RMARTS Project members' considerable capability as change agents. Each year's plan presents material on transition action to be taken. In the 1990 and 1991 plans, sep-

arate sections appear for "technology insertion/adoption tasks" and for "transition plans." Tasks include technical work, but also highly leveraged transition work, such as helping with the Navy's Next Generation Computer Resource (NGCR) standards development and finding vendors for commercial distribution, training, and consulting.

6.2 Standards Work

Standards efforts represent a high-leverage activity for improving computing and software engineering since they are community efforts, developed and distributed by organizations such as IEEE (Institute of Electrical and Electronic Engineers), ANSI (American National Standards Institute), NIST (National Institute of Standards and Technology) and ISO (International Organization for Standardization).[24] Standards take years to reach official approval with multiple intermediate drafts circulated for comments and voting by the technical community; however, RMARTS recognized the importance of contributing to precompetitive consensus building and to standards efforts.

Using standards to transition technology allows one to take advantage of mechanisms for dissemination that are already in place. Craig Meyers, of RMARTS, comments: "When we work on an IEEE standard, IEEE takes care of publishing the standard, and they're the ones who route it to ANSI and ISO. The preexistence of mechanisms to advertise, publish, and disseminate make transition much easier for us. And when vendors whose products conform to the standards begin to advertise product features using the terminology of the technology, we get more free leverage."[25]

Standards that might potentially block a technology need to be modified to permit or, ideally, to support the adoption of that technology. Activities throughout the life of RMARTS demonstrate understanding of standards as a means of reducing barriers to technology adoption. In the case of RMA, multiple standards efforts were pursued (see Appendix A).[26] As early as 1988, RMARTS began to work with hardware bus standards groups (IEEE P896.3 FutureBus+) to allow effective use of prioritized scheduling algorithms. These efforts continued through 1988-1989; and in 1990, two related papers were published. These papers addressed

[24] Formal standards cannot be overestimated as a high-leverage transition activity. IEEE tracks its standards to become US national standards through ANSI, which puts them on the path to becoming international standards through ISO. Working with IEEE standards, then, leads to standardization in the international marketplace, the broadest possible arena of influence.

[25] Personal interview with RMARTS Project member Craig Meyers, conducted by Bill Pollak, July 1993.

[26] Ted Baker, resident affiliate with RMARTS from 1991-92, works with three standards efforts: Ada9x Mapping/Revision Team, POSIX (portable operating system interface) working group (WG) IEEE P1003.4 (developing standards for real-time operating system services), and POSIX WG IEEE P1003.5 (developing standard Ada language bindings for the POSIX standards). Since 1991, Michael González Harbour of the Universidad de Cantabria, visiting scientist with RMARTS from 1991-1992, has also been active with POSIX WG IEEE P1003.4. John Goodenough has been a key contributor to development of the Ada9X programming language standard. Craig Meyers is chair of a working group that is writing a POSIX standard (IEEE P1003.21) for real-time distributed systems communication. In addition, Meyers has been active in the SAFENET (survivable adaptable fiber optic embedded network) Working Group and IEEE P1003.12, Protocol Independent Interfaces for Interprocess Communication Working Group. Meyers and Lui Sha participated in the NGCR Real-Time Working Group (focused on system integration). Sha has led and been active in P896.3 standardization efforts for IEEE FutureBus+.

design issues concerning support for real-time system development based on rate monotonic scheduling and the IEEE FutureBus+. Also in 1990, the *System Design Manual* for IEEE FutureBus+ included a chapter on how to use the FutureBus+ when designing real-time systems based on RMA. Particularly interesting, here, is the way that the RMARTS team used convergent vehicles for transition. For example, the standards work was, initially, a mechanism for getting the technology adopted; it was later realized that the standardization process itself was also a mechanism for disseminating technical information and its utility: first, by explaining it to the members of the standardization group and convincing them to use the theory in creating a standard; and second, by increasing interest in the theory when people heard about the theory's presence in the standard.

FutureBus+ was not the only standards effort. In 1990, RMARTS members were active in the Navy's Next Generation Computing Resources (NGCR) SAFENET (survivable adaptable fiber-optic embedded network) Working Group and the Real-Time Working Group, also under NGCR, which was focused on system integration. 1992 activities included participation with two other portable operating system interface (POSIX) efforts: Real-Time Distributed System Communication (IEEE P1003.21) and Protocol Independent Interfaces for Interprocess Communication (IEEE P1003.12). Standards efforts also included work on Ada language maintenance and revision (1988); and activity to influence the Ada9X programming language standard to encourage use of RMA algorithms (begun in 1990). (References to the complete set of standards activities are provided in Appendix A.)

Of the total of 132 activities coded, 15 (about 11%) were related to standards, including IEEE FutureBus+, POSIX, and Ada 9X.

The catalogue of standards activity alone does not capture the strategic intent of the RMARTS Project—to integrate standards into the vision. Goodenough summarizes this intent here:

> For RMA, you couldn't get these scheduling concepts actually used until you permeated all the places in which they were needed, which meant having them supported, or at least not blocked, by the programming language, the operating system, and the local area network.
>
> If you want to get the technology into use, you have to ensure that the relevant standards that might block the technology are modified to permit it or ideally, to support it. In the beginning we found lots of people saying, "Well, that's a great idea, but my operating system doesn't support it; what can I do about it"? That's when we began to focus on how people could use RMA even if their operating system didn't support it. That was one of the reasons for dealing with standards right in the beginning.

> One of the first things we did was say, "How can we interpret the Ada standard to allow an operating system to conform to the standard and still to support RMA scheduling?" In fact, some of my first presentations on RMA were to people involved in the ADA Language Maintenance Committee so that they would be sympathetic to viewing this broader interpretation. Provoking the issues in a standards context caused people to pay much more attention so that we could persuade them.
>
> The fact that RMA concepts appeared in the Ada9X standard and in the POSIX standards gave them an inherent credibility. It meant that these ideas had been debated among industry people who aren't interested in any ideas unless they actually help them out. The fact that these ideas were being discussed in a standards committee, which is generally composed of people who are influential within their companies and within their fields, meant that these ideas were reaching key opinion makers. It's like targeting your technology to the people who will make the most difference. When other people see them in the standard, they begin to say, "Gee, what is this stuff? I ought to know about it because it's passed this hurdle of being accepted." So, in terms of generating awareness of a technology, standards are an excellent vehicle.[27]

A final point about standards activity is important. Many readers of this case study will be working in business enterprises where technology development and transfer are directly related to market share and profit motive. The SEI has neither such relationship; and the lack of this relationship had a direct and positive effect on the RMA success with standards organizations. The point is worth underscoring for two reasons: first, the SEI occupies a special and unusual position; second, in order to follow the RMA example, commercial organizations will have to overcome the assumption that they are motivated solely by profit.[28]

6.3 Range and Pattern of Transition Activity

In any particular year of the RMA transition chronology, one or two themes dominate. For example, in 1987 most of the activities address key barriers to adoption of the technology; and in 1988 technical partnerships and collaboration are a dominant theme. Together, the variety of transition activities for the first two years of RMARTS is unusual. In the following brief discussion, we comment on the range of transition activities and important patterns that we see, from 1987 through 1993.

---

27. Personal interview with John Goodenough conducted by Bill Pollak, July 1993.

28. Personal conversation with Tom Ralya, November 1993.

In 1987, while the main thrust of activities addresses the technical limitations, IBM is deploying hardware/software with periodic-only rate monotonic scheduling (RMS). In 1988, two resident affiliates, one from Naval Weapons Center and one from IBM, are on board; the IBM affiliate, collaborating with senior engineers back home, is preparing material for an IBM internal course. Also in 1988, several compiler vendors are experimenting with rate monotonic scheduling algorithms; and RMARTS has started to work with a hardware bus standards group to allow effective use of prioritized scheduling algorithms.

In 1988 and 1989, collaboration with a single large organization, IBM, accounts for roughly 25% of total activities. Most significant are the arrival of the IBM resident affiliate, preparation of an IBM internal course by the affiliate, development of the generic avionics case study (with the Naval Weapons Center resident affiliate), and application of RMA to the BSY-1 Team Trainer system. In interviews, Goodenough indicated that the choice of one large influential organization was part of the transition strategy for RMA, the idea being to establish credibility by association as well as to take advantage of the substantial engineering expertise offered by the organization.

By 1990, several organizations are actively involved with the project; IBM continues, but now the Navy, General Electric, and the Space Station Project (including contractors such as McDonnell Douglas, Honeywell, and the Research Institute for Computing and Information Systems [RICIS]) are also engaged. Again, the strategy is to use the influence of the large organizations to draw attention to RMA. Also in 1990, the European Space Agency (ESA) On-Board Data Division announces plans to use RMA as the baseline methodology for its hard real-time operating systems project. While the source of the connection to the ESA is not known, its plans signal the extent to which word about RMA is spreading. This may indicate that the "influence" strategy is working.

Standards work, discussed at length above, which began in 1988 and 1989, picks up steam in 1990 and 1991. The reference to an engineering handbook first appears in 1991. The handbook is worked on intensively in 1992, culminating in publication by Kluwer Academic Publishers in 1993 of *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems.*[29]

Also in 1990 and 1991, Navy support for work with General Electric[30] provides extensive opportunity to understand and refine the introduction process for the technology. This includes development of an RMA workshop approach that gives students hands-on experience applying the method, and development of data sheets—templates for collecting information about each task that is needed to perform a timing analysis. A modification of the data sheet approach was incorporated into the handbook later, in 1992.

---

[29] Goodenough notes that the idea of a handbook appeared much earlier, in a draft plan that he prepared in 1987. He had a "long-term goal of putting the output of projects into handbooks." Personal correspondence with John Goodenough, October 1993.

[30] RMARTS Project member Mike Gagliardi's engineering background was essential in his work with GE; he had credibility as a practicing engineer and could answer questions from that viewpoint. He was seen by his students as one of them. Moreover, he had worked on the GE project and was recognized as a trustworthy authority.

Whereas technical partnerships were key in 1988, partnerships involving distribution are common in 1992. These partnerships focus on diffusion and creation of the whole product [Moore 1991] for RMA—tools, commercial training, courses, videotapes, and the RMA handbook. For example, students from the Carnegie Mellon University Masters in Software Engineering Program collaborate with RMARTS to develop a real-time analysis tool. Kluwer is engaged to publish the handbook. Companies that give technology training courses are pursued to encourage them to give RMA training courses. Telos and Tri-Pacific were eventually selected and signed agreements with the SEI enabling the development of commercial courses on RMA (which were then listed in SEI's Products and Services portfolio). These two companies also sponsor booths at the SEI Software Engineering Symposium. Ruth Ravenel of the University of Colorado begins development of a short video-based course, with project cooperation. An agreement with the Software Productivity Consortium (SPC) is negotiated to allow incorporation of RMA into the Software Productivity Consortium (SPC) ADARTS™ methodology. The user community, at the instigation of the SEI and as part of the RMARTS self-sustaining transition strategy, begins to be organized through the first RMA Users Forum in September 1992.

Finally, several general observations are in order. Throughout, RMARTS members were attentive to the notion of "leveraging the infrastructure," an idea frequently discussed at the SEI during its first few years. As we have indicated, the idea was to expedite "self-sustaining transition" of RMA so that project members could discontinue interventions and move along to determining and developing the next "promising technology." The decision to end the project was made 18 months before the scheduled termination of the project, and the stage model was explicitly used to focus efforts during this 18-month period on achieving self-sustaining transition. High-leverage activities, such as partnerships for co-development and distribution, were selected and chosen with this goal in mind. Hence the focus on training partners and on capturing existing knowledge about RMA in a handbook. Both RMARTS and its partners benefited from the arrangement. More examples of this can be seen in the "self-sustaining transition" phase in Appendix A.

Project members understood the need to reduce risk for potential partners. For example, in the case of workshops in the application of RMA, RMARTS developed material and delivered sufficient offerings themselves to understand the type of training needed. They did not, however, provide ongoing delivery for the course, but recruited (or supported volunteer) training groups such as the IBM corporate education organization, RICIS training for NASA personnel, and Telos and TriPacific as commercial suppliers. In contrast, tool development has progressed much more slowly, perhaps because there are more work and resources involved in taking an SEI-developed prototype tool and creating a commercial-grade product than in taking training materials and upgrading them to a commercial-quality course.[31]

---

31. "The work on tool development was limited to the development of a calculation program that was distributed to interested parties on request with the advice that it was only an example of what could be done." In retrospect, Goodenough comments: "We explicitly decided not to focus on tool development because other areas seemed to have higher leverage. Since we didn't have the resources, we hoped that by staying back, some commercial vendor might jump in and provide a tool. But if we had had more resources, we might have provided a tool." Personal correspondence with John Goodenough, October 1993.

In reviewing transition activities from 1987 to1993, we are able to observe how the type of transition activity changed during the life of the project. Over time, effort spent on extending the technology itself diminished and effort expended on demonstrating the usefulness of the technology increased. Similarly, in the later years, more and more effort was dedicated to "leveraging the infrastructure" and disseminating the technology to the broader population of candidate users. By 1993, SEI involvement was limited to transition-specific activities such as the RMA Users Forum and *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems.* At this time, the set of transition activities has migrated out of RMARTS and into Products and Services Planning. Figure 6-1 summarizes a six and one-half year chronology of activity related to the transition of RMA.[32] Each unit in each of the cells of the matrix represents a significant project activity, such as the arrival of a resident affiliate at the SEI or the publication of a technical paper.

[32] In Section 5, Method, Figure 5-1 shows complete information for 1988; Appendix A includes details for all years.

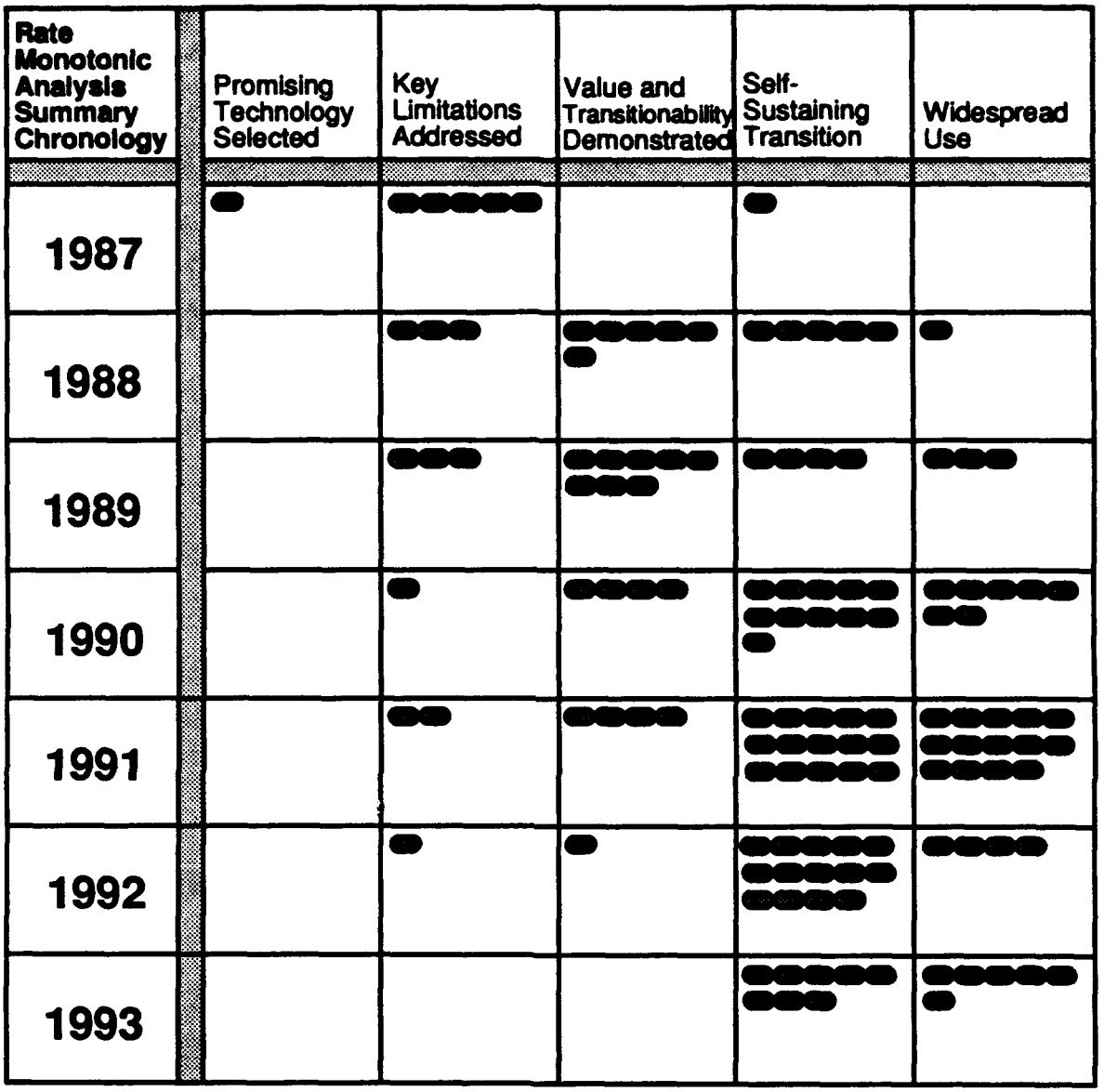


Figure 6-1: Summary of RMA Activities

A number of interesting patterns appear when the data are viewed in this summary form. First, the level of effort increased in each year of the RMARTS Project through 1991. Second, in each year, activities occurred in at least three of the five life cycle phases, belying any assumptions about linearity of progress across the phases. In this regard, we note that the RMARTS transition model and our manner of coding have allowed us to override limitations, such as linearity, typically associated with stage models of technology development.

A crude estimate of activity, assuming each unit is approximately equivalent, shows total activity nearly doubling from "value and transitionability demonstrated" to the "self-sustaining" phase; and the total remaining fairly steady from the "self-sustaining" to "widespread use"

phases. Viewed chronologically, number of activities per year increased steadily until, in 1991, number of activities was about five times that of 1987. In 1992 and 1993 the activity level dropped off. This is because, as we have noted, at the end of 1992 RMARTS closes down, maintaining that self-sustaining transition had been reached; and in 1993, RMA transition support continued through Products and Services Planning. Given the limited resources of the SEI and its concern with leverage, the smaller number of activities in the "widespread use" phase is not surprising.

Figure 6-2 presents the total count of transition activities, year by year.

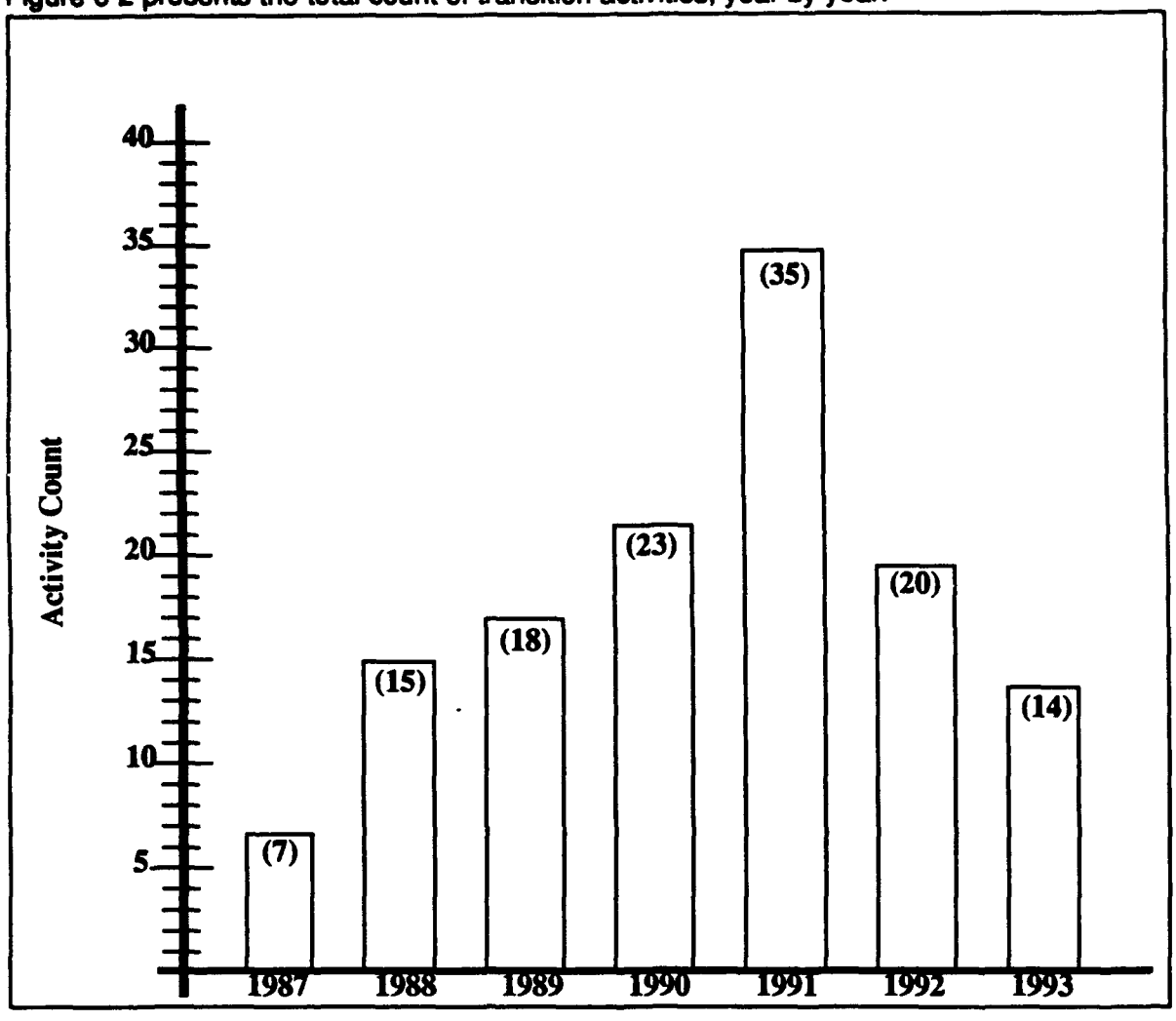


Figure 6-2: Transition Activity Counts by Year

This is a somewhat different representation of the information conveyed in Figure 6-1, the summary of activities. It is important to note that at the same time SEI activities began to fall off, in 1992 and 1993, activities on the part of external agents (beyond the scope of the present study) began to rise. For example, in 1992 Telos and Tri-Pacific began investing their own resources to develop commercial training for RMA. Similarly, in 1993, Kluwer was contributing resources to the production of the RMA handbook.

A comment on the shape of Figure 6-2 is in order. Of interest is the resemblance between the curve here and the standard, commonly held, adopter population profile, including innovators, early adopters, early majority, late majority, and laggards. [Rogers1983]. We cannot elaborate on external activities for 1992 and beyond to fully account for the (development-related) adopter population for RMA, and the relationship between this curve and the adopter-population profile is speculative; however, the similarity suggests a provocative line of inquiry. Technology developers, co-developers, and collaborators may well share some of the characteristics typically associated with adopters or consumers—the less mature the technology, the greater the need for partners who are innovators or early adopters [Rogers 1983].

Finally, Figure 6-3 provides an overall picture of how primary transition activities related to each other over time. This representation was constructed by RMARTS member Tom Ralya and Mario Barbacci, manager of the SEI Real-Time Systems Program, the program in which RMARTS was located.

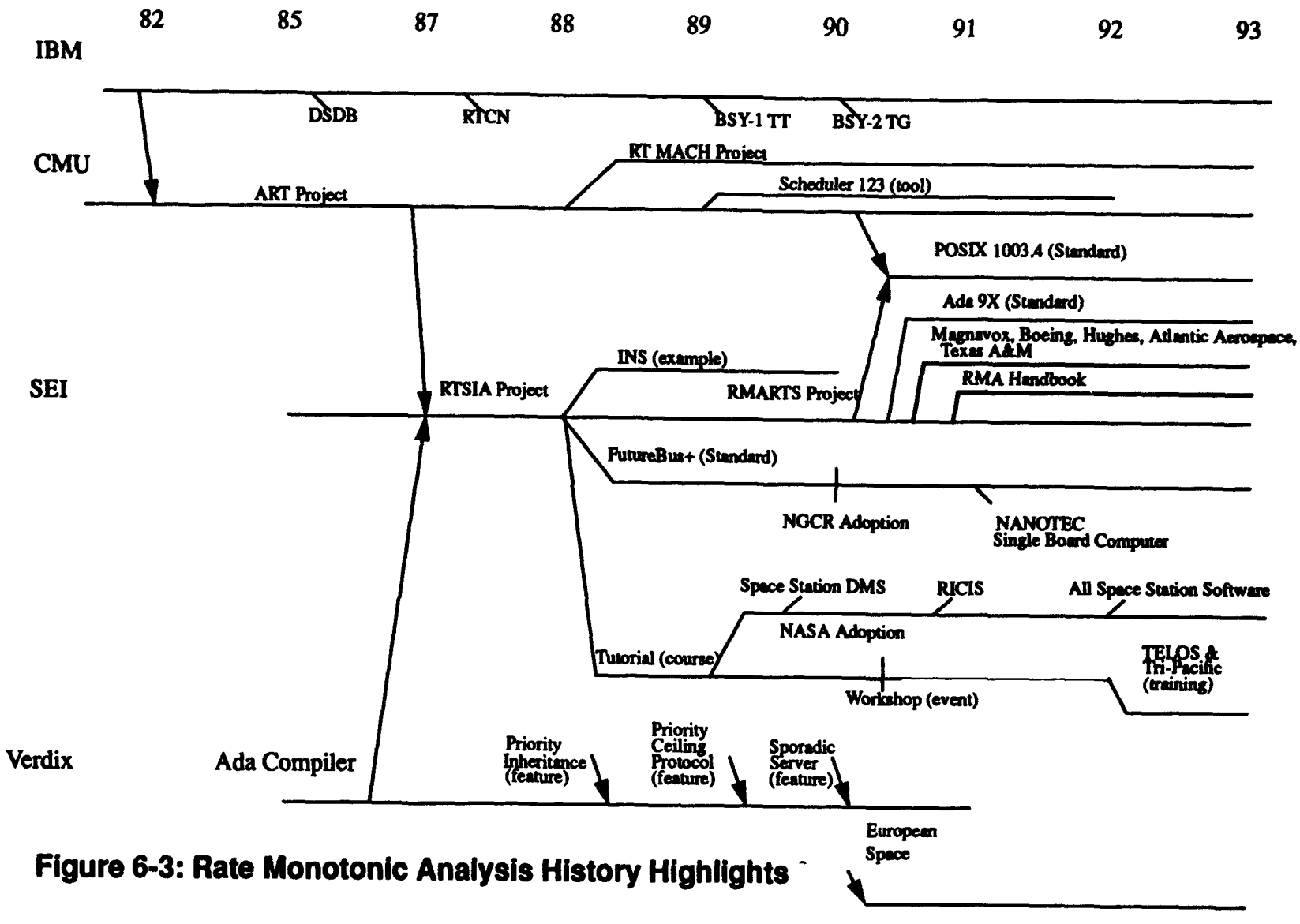


Figure 6-3: Rate Monotonic Analysis History Highlights

7 Discussion

What can be learned from the RMA case study? As might be expected, the results of the study have implications for a range of individuals and the different types of organizations with which they are affiliated. In the following *Discussion* section, we comment briefly on implications for: developers; managers of R & D, including funding agents; and researchers. We offer a final observation on the contribution that research, especially the case study, makes to a greater understanding of technology transition.

7.1 Implications for Developers of Maturing Technologies

Key issues in this area include:

- The mix of skills and benefits of an interdisciplinary team.
- Variety of transition mechanisms.
- Project vision.
- Partnerships for development and distribution.

Technologists working with less mature or maturing technologies must use a variety of skills. The RMARTS Project's use of different types of technical people, including a resident affiliate from an influential organization, academics, engineers, and mathematicians fostered the crossing of engineering-related boundaries.

A larger gap existed between those in technical disciplines and those in "softer" disciplines, such as marketing, management, technical communication, and training; and within the SEI, allocation of such resources was problematic. "It was difficult to secure upper management approval for people to spend sufficient time on RMARTS... [W]e tried to be interdisciplinary early on, but the organization wasn't structured in a way that let us be helped."[33] RMA transition was expedited because of this eventual collaboration; we can only speculate about the benefits of constituting the interdisciplinary group even earlier on.

The mix of individual skills and backgrounds was complemented by a variety of transition mechanisms. Technical papers and tutorials, quite conventional mechanisms, were used by the RMARTS Project. Early on, they also used unconventional mechanisms: standards work, resident affiliates, partnerships with compiler vendors, and test cases worked with engineers in external organizations.

If transition is a goal early in the development of a technology, concern about transition success pervades the development effort. RMARTS personnel had early exposure to SEI transition concepts and goals. We have seen how Goodenough and Sha incorporated transition into their project vision through the RMARTS stage model. While the need for a vision may strike

---

[33] Personal correspondence with John Goodenough, October 1993.

some as obvious, many technology developers are unfamiliar with strategic planning or third-generation R & D management [Roussel, Saad, & Erickson 1991].

Once the vision is developed, a transition strategy must follow. This requires an understanding of how the technology being developed will be used. In 1988 and 1989, RMARTS personnel worked with engineers outside the SEI and learned early about what roadblocks would occur when people tried to adopt RMA to build "real" systems. Knowledge of these obstacles was subsequently translated into questions and actions:

- How would compilers have to change?
- Would engineers see RMA as applicable to the design of new systems and to the tuning and upgrading of old systems?
- What would engineering managers need to know?

Partnerships were critical. When an organization is new, small, or both, there is great advantage in teaming with a larger partner, preferably one that is influential and has deep pockets. RMARTS chose such a partner in IBM,[34] gaining not only resources such as a resident affiliate and the working of case examples, but also gaining considerable credibility. In addition to early co-development partnerships, distribution partnerships were necessary: RMARTS may have engaged third-party vendors for training and publication fairly late in the technology development life cycle. In hindsight, project members observed that partnerships with trainers and tool vendors might have been worked earlier on in the life cycle. We cannot say to what extent the transition of the technology would have been accelerated by focusing on new product development concurrently, alongside technical development and extension.[35] An increasing number of indicators argue in favor of mutual adaptation of technology development and implementation [Leonard-Barton 1988a].

Finally, developers and practitioners must continue to reflect upon their transition activities and find efficient ways to capture their processes so that lessons can be shared with others through experience reports. They must consider:

- What aspects of the process are important?
- How best can they be described?
- To whom are these descriptions relevant?
- How early in the maturation of a technology can a transition process be captured and still predict later transition approaches?
- How can we anticipate the needs of the change agents who must introduce the technology under consideration?

---

[34] It is important to remember that IBM had a longstanding relationship with the Advanced Real-Time Technology (ART) Project at Carnegie Mellon, beginning in 1982 and continuing until the present.

[35] This is a problematic issue. Evidence indicates that accelerating technology transition requires focusing upon technology development and product development concurrently. However, if there is no clear existing or potential market, it is difficult to engage the interest of product developers early in the technology life cycle. The question remains: what would make product developers take the leap of faith that they needed to invest earlier than they did?

These are difficult questions, requiring more than generic answers. Developers and practitioners may want to turn to case study research as well as others' experience reports for guidance in answering these questions.

7.2 Implications for R & D Managers and Funding Agents

Process-intensive technologies such as software, including methods for designing, developing and testing software such as RMA, require implementation-intensive transition methods. While there is evidence that software technology transition can be more systematic [Leonard-Barton 1988a, Leonard-Barton 1988b] and approaches to transition can be replicated [Ackerman 1983, Bouldin 1989, Grady & Caswell 1990], the problem of "discovering" an approach in the first place remains. In R & D, during the construction of prototypes, if attention is given to adoptability and ease of use as well as to technical correctness, then information can be obtained that is helpful to the design of commercial-grade products or techniques. These commercial products are "whole" [Moore 1991]; they incorporate the secondary products and services that late majority adopters need: training and support, courses, documentation, handbooks, etc.

Those funding technology development within their own organizations, or in external organizations such as universities, expect that the technology being developed will automatically be used. While advice for technologists and managers on processes for selecting an optimal mix of R & D projects is available—for example, [Roussel, Saad, & Erickson 1991]—guidance on how to attend to the transfer of specific types of technologies that these projects may develop is rare. In addition, software technologies require a *transition*—not just a binary transfer—process. Funders need to understand what development and/or transition activities they are funding: where in the value chain or maturation life cycle the technology is, what arena the technology is intended for, and what the relative cost will be. (They will also benefit by attending to the issues for developers described above.) The example of RMA transition does not provide all the answers, but the level of detail about types of activities over a significant time period may stimulate thinking about requirements for other technologies. These requirements might concern: staffing, facilities, schedule, and financing.

7.3 Implications for Transition Research

The case study on the transition of RMA represents a single effort to understand the complex processes of software technology transition and diffusion. In order to gain a full understanding of this area, researchers must consider both sides of the technology push/technology pull equation and attend to the full range of transfer conditions from R & D, through new product development, to adoption and implementation in an organizational setting. Such an understanding of technology transition requires crossing boundaries between disciplinary communities and between the arenas of academia, industry, and government. The latter is a difficult task: each discipline and each arena speaks about and investigates technology transfer in a different way [Fowler & Levine 1993, Rogers 1983, Tornatzky & Fleischer 1990].

Additional case study research on technology transition in general and software technology transition in particular is essential. Researchers must continue to explore the circumstances and surrounding conditions for transferring software tools, techniques, or practices, as opposed to other kinds of technologies. Such distinctions will become critical as more and more technology is layered—and as software is embedded within other technologies. Within the software transition arena, researchers must begin to focus on a taxonomy of technology types and consider distinctions between tools, techniques, integrated toolsets, and larger ill-defined process-based technologies. Because of its nature, software raises fundamental questions about process- and product-based technology. Understanding these relationships will help create new taxonomies for technology beyond current manufacturing operation distinctions, such as customer technology, small and large batch, mass production, etc. [Woodward 1965, Khandwalla 1974]

Case study research can provide the basis for understanding software technology transition issues generally, and for exploring transition with respect to specific technology types and organizational settings. How does the transition of RMA compare to that of a CASE tool? A software development environment? A human behavior-intensive technology such as software inspections? A tool-based technology such as project management that also requires human behavioral changes in attention to detail? A grand-scale composite technology such as software process improvement? Finally, how does the transfer of these technologies vary with respect to the nature of the receiving organization or environment?

In exploring these and other questions, researchers must be willing to innovate and to borrow methods of inquiry from the many disciplines that shed light on the process of technology transition. For example, the study of RMA raises interesting questions about the types of people involved and the communities that they represented. A full history and communication-network analysis might provide many clues to issues of diffusion and influence.

Case study research offers us a way to understand, in depth, how software technologies can be effectively (or not effectively) transitioned. Such studies support the goals of technology developers, their managers and funding sources, and researchers alike. They reflect and honor the complexity of real transition situations, and also, as their number increases, provide a basis for common understanding of a range of transition experience for software technologies.

Acknowledgments

We would like to acknowledge the assistance of John Goodenough, RMARTS Project Leader, and RMARTS Project members: Ted Baker, Mike Gagliardi, Michael González Harbour, Mark Klein, Craig Meyers, Tom Ralya, and Lui Sha. Others provided helpful advice along the way: Larry Druffel, Director of the SEI; Mario Barbacci of the Real-Time Systems Program; Jane Siegel of Empirical Methods; Ray Obenza of Products and Services Planning, Lynn Robert Carter of Technical Assistance, and Neil Eastman of Motorola.

We thank Pam Hughes for help preparing the manuscript, and Bill Pollak for editorial assistance. Finally, special thanks are due to Tom Ralya for sharing his insights on the technology and the RMARTS Project from the perspectives of insider and outsider, as project member and resident affiliate.

References

Adler, P.S. & Shenhar, A. (1990). Adapting your technological base: The organizational challenge. *Sloan Management Review 32*(1), 25-37.

Botkin, J. & Matthews, J. (1992). *Winning combinations: The coming wave of entrepreneurial partnerships between large and small companies.* New York: John Wiley & Sons, Inc.

Bouldin, B. (1989). *Agents of change: Managing the introduction of automated tools.* Englewood Cliffs, N.J.: Yourdon Press.

Chadwick, B.A., Bahr, H.M., & Albrecht, S.L. (1984). *Social science research methods.* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Conner, D.R. & Patterson, R.W. (1982). Building commitment to organizational change. *Training and Development Journal*, April 1982, 18-30.

Deal, T.E. (1991). Developing a quality culture. In R.H. Kilmann (Ed.), *Making organizations competitive* (pp.156-175). San Francisco, Calif.: Jossey-Bass.

Dean, J.J. (1987). *Deciding to innovate: How firms justify advanced technology.* Cambridge, Mass.: Ballinger Publishing Company.

Downs, G.W. & Mohr, L.B. (1976). Conceptual issues in the study of innovation. *Administrative Science Quarterly 21*, 700-714.

Fagan, M. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal 15*(3), 182-211.

Fichman, R.G. & Kemerer, C. (1993). Adoption of software engineering innovations: The case of object orientation. *Sloan Management Review*, Winter 1993, 7-22.

Fowler, P. & Levine, L. (1992a). Software technology transition: Life cycles, models, and mechanisms. Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute, *Bridge* (October), 3-8.

Fowler, P. & Levine, L. (1992b). Toward a problem solving approach to software technology transition. In J. Van Leeuwen (Ed.), *Proceedings of the IFIP 12th World Computer Congress, vol. 1* (pp. 57-64), Madrid, Spain. The Netherlands: North Holland, Elsevier Science Publishers.

Fowler, P. & Levine, L. (1994). From theory to practice: Technology transition at the SEI. In *Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii. IEEE Computer Society Press.

Fowler, P. & Maher, J. (1992). *Foundations for systematic software technology transition.* Software Engineering Institute Technical Review '92, 1-32.

Fowler, P. (1990). Technology transfer as collaboration: The receptor group. *Proceedings of the 12th International Conference on Software Engineering* (pp. 332-333), Nice, France. U.S.: IEEE Computer Society Press.

Geertz, C. (1973). Thick description: Toward an interpretive theory of culture. In *The interpretation of cultures: Selected essays.* New York: Basic Books, Inc.

Glynn, G. (1990). Semi-formal process model for technology transfer. *Proceedings of the 12th International Conference on Software Engineering* (pp.334-335), Nice, France. U.S.: IEEE Computer Society Press.

Grady, R.B. & Caswell, D.L. (1987). *Software metrics: Establishing a company-wide program.* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Humphrey, W.S. (1988). Characterizing the software process, *IEEE Software 5*, 73-79.

Kanter, R. M. (1983). *The change masters.* New York: Simon and Shuster.

Khandwalla, P.N. (1974). Mass output orientation of operations technology and organizational structure. *Administrative Science Quarterly 19*, 74-97.

Klein M. et al. (1993). *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems.* Boston, Mass.: Kluwer Academic Publishers.

Leonard-Barton, D. (1988a). Implementation as mutual adaptation of technology and organization. *Research Policy 17*, 5 (Oct. 1988), 102-110.

Leonard-Barton, D. (1988b). Implementation characteristics of organizational innovations. *Communication Research 15*, 603-631.

Leonard-Barton, D. The intraorganizational environment: Point-to-point versus diffusion. In F. Williams & D.V. Gibson (Ed.), *Technology transfer: A communication perspective.* Newbury Park, Calif.: Sage Publications, 1990.

Liu, C.L. & Layland, J.W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the Association for Computing Machinery 20,* 46-61.

Moore, G. (1991). *Crossing the chasm: Marketing and selling technology products to mainstream customers.* Harper Business.

Morgan, G. (1986). *Images of organization.* Beverly Hills, Calif.: Sage Publications.

Nord, W.R. & Tucker, S. (1987). *Implementing routine and radical innovations.* Lexington, Mass.: D.C. Heath.

Paulk, M. et al. (1991). *Capability maturity model for software.* (Technical Report CMU/SEI-91-TR-24; ADA-240603). Pittsburgh, Pa.: Software Engineering Institute.

Pollak, W. (1993). Standards: Leverage for new technologies. Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute, *Bridge* (October), 1-7.

Przybylinski, S., Fowler, P., & Maher, J. (1991, May). *Software technology transition.* Tutorial presented at the 13th International Conference on Software Engineering, Austin, Texas.

Redwine, S.T., Becker, L.G., Marmor-Squires, A.B., Martin, R.J., Nash, S.H., & Riddle, W.E. (1984). *DoD related software technology requirements, practices, and prospects for the future.* (Technical Report IDA Paper P-1788). Alexandria, Va.: Institute for Defense Analysis.

Roberts, E. (1991). *Entrepreneurs in high technology: Lessons from MIT and beyond.* New York: Oxford University Press.

Rogers, E.M. (1983). *Diffusion of innovation.* New York: Free Press.

Roussel, P.A., Saad, K.N., & Erickson, T.J. (1991). *Third generation R & D: Managing the link to corporate strategy.* Boston: Harvard Business School Press.

Sha, L. & Goodenough, J.B. (1990). Real-time scheduling theory and Ada. *IEEE Computer 23*(4): 53-62, April 1990.

Sha, L., Klein, M.H. & Goodenough, J.B. (1991) *Rate monotonic analysis for real-time systems* (Technical Report CMU/SEI-91-TR-6, ADA235641). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University.

Souder, W.E. (1987). *Managing new product innovations.* New York: Lexington Books.

Souder, W.E., Nashar, A.S., & Padmanabhan, V. (1990). A guide to the best technology transfer practices. *Journal of Technology Transfer 15*, 5-16.

Tornatzky, L.G. & Fleischer, M. (1990). *The processes of technological innovation.* Lexington, Mass.: Lexington Books.

Walton, R.E. (1989). *Up and running.* Boston: Harvard Business School Press.

Williams, F. & Gibson, D.V. (Eds.). (1990). *Technology transfer: A communication perspective.* Newbury Park, Calif.: Sage Publications.

Willis, R. (1983). Technology transfer takes 6 +/- 2 years. In R. Morton (Ed.), *Proceedings of the IEEE Computer Society Workshop on Software Engineering Technology Transfer* (pp.108-117), Miami Beach, Fla. IEEE Computer Society Press.

Woodward, J. (1965). *Industrial organization: Theory and practice.* London: Oxford University Press.

Yin, R.K. (1984). *Case study research: Design and methods.* Beverly Hills, Calif.: Sage Publications.

Zmud, R.W. & Apple, L.E. (1992). Measuring technology incorporation/infusion. *Journal of Product Innovation Management 9*, 148-155.

Appendix A Rate Monotonic Analysis 1987-1993

1987

(Project was known as Real-Time Scheduling in Ada--RTSIA)

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
Theory of rate monotonic scheduling (RMS) obtained from CMU.	Theory used at IBM Manassas for BSY-1 real-time communication network (DSDB) 82-87 and on R&D project (RTCN) 86-87. Conference paper on priority ceiling protocol. Work with compiler vendors to implement proposed scheduling strategies. Real-time systems examples and test cases from Navy, IBM. Priority inversion problem in Ada tasking and solution accepted by 1st Int'l Workshop on Real-Time Ada. This leads to Verdix providing Ada compiler runtime source code for experimentation.		IBM deploys HW/SW with periodic-only RMS.	

1988

(Project was known as Real-Time Scheduling in Ada)

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Reports for compiler vendors about implementing algorithms for Ada runtime systems. Development of theory implementation validation and performance tests. Analysis of Real-Time Embedded Systems Testbed (REST) Project's Inertial Navigation System for refinement of theory.	IBM resident affiliate arrives. Naval Weapons Center (NWC) resident affiliate arrives. IBM resident affiliate in collaboration w/ senior IBM engineers modifies tutorial materials for IBM-internal RMS course. Generic avionics case study developed by IBM and NWC. Naval Weapons Center resident affiliate developing generic missile case study. Representation of the avionics system implemented using modified Verdix Ada compiler to determine if predicted performance can be obtained.	Work with hardware bus standards groups (IEEE FutureBus+) to allow effective use of prioritized scheduling algorithms. Verdix implements RMS-required scheduling algorithms in its Ada runtime system. Other compiler vendors experimenting with the algorithms. Work with Ada language maintenance and revision efforts. IBM experiment with RTCN at IBM.	One-day tutorial prepared to give to engineers at SEI Affiliates Symposium.

1989

(Project was known as Real-Time Scheduling in Ada)

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Analysis of REST Project's Inertial Navigation System (INS) for refinement of theory completed and modifications made to both theory and INS system. Continued interaction with related projects in Carnegie Mellon University's School of Computer Science that focus primarily on scheduling theory. An Ada program that can evaluate schedulability algorithm is developed.	Naval Weapons Center resident affiliate continues developing generic missile case study and consulting on development of a generic avionics application. IBM resident affiliate consulting on application of RMS theory to distributed systems. IBM demonstrates value of RMA internally by applying it to BSY-1 Team Trainer. IBM (at IBM) develops a generic avionics application in cooperation with NWC to demo application of RMS theories. IBM loans RTCN to SEI for broader experiment. IBM demonstrates value of RMA by applying it to BSY-2 Transmit Group. Representation of the avionics system implemented using modified Verdix compiler to determine if predicted performance can be obtained. Work with Navy's Next Generation Computer Resources (NGCR) Program to encourage development of a LAN that supports RMA principles. Issues in the transition of RMA to a distributed system are identified. As a result of this effort, a serious priority inversion in 802.5 is uncovered.	IBM incorporates RMS into internal real-time systems training course, based largely on SEI tutorial material. Hardware bus standards groups (IEEE FutureBus+) continue to develop standards that allow effective use of prioritized scheduling algorithms. Verdix continues implementing RMS-required scheduling algorithms in its Ada runtime system. Other compiler vendors continue experimenting with the algorithms.	One-day tutorial given to engineers at SEI Affiliates Symposium. One-day tutorial offered at TriAda89 breaks tutorial attendance records with est. 220 attendees (usually about 40). Verdix has included changes incorporating RMS in recent compiler releases.

1990

(Project was known as Rate Monotonic Analysis for Real-Time Systems--RMARTS)

NOTE: Focus has shifted from scheduling algorithms to broader application of RMS theory to design and analysis of real-time systems. Also, name change is due to RMS theory becoming more widely known and thus usefully mentioned in the title of the project. Ada remains a preferred mechanism for implementing RMS theory but is not the primary focus. And because rate monotonic analysis is now the focus, the acronym is RMA.

Promising Technology Selected	**Key Limitations Addressed**	**Value and Transitionability Demonstrated**	**Self-Sustaining Transition**	**Widespread Use**
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Technical report, *An I/O Paradigm for Real-Time Systems*, shows how RMA can be used as a factor in determining design tradeoffs.	Inertial Navigation System (INS) analysis demonstrates applicability of RMA. A generic avionics case study demonstrates a new design for such systems based on RMA principles (continues 1989 work). A paper ("A Real-Time Locking Protocol") extending the rate monotonic approach to address the locking problem in real-time database applications accepted by *IEEE Transactions on Computers*. BSY-1 trainer study shows how RMA can lead to dramatic improvements in an existing system.	Benchmark tests for Ada compilers to check implementation of RMS algorithms are developed and beta tested. IBM continues to offer RMA in internal real-time systems training courses. SEI technical report, *Implementing Sporadic Servers in Ada*, is published, demonstrating to Ada programmers and compiler vendors how the sporadic server algorithm can be supported in Ada. First in a series of SEI technical reports on how RMA addresses common real-time programming issues, *An Analysis of Input/Output Paradigms for Real-Time Systems*, is published. SEI teaches RMA course to various BSY-2 subcontractors and prime contractor. Prime contractor develops its own course and begins teaching RMA internally and to other subcontractors.	RMA tutorial/workshop offered at NASA Johnson Space Center and at McDonnell Douglas (prime contractor for the space station). NASA adopts RMA for the space station data mgmt system. McDonnell Douglas and subcontractors such as IBM and Honeywell agree to use RMA as the baseline approach for designing real-time software. NASA considers requiring use of RMA as baseline approach for all real-time software. RMARTS works with GE on various subsystems of the BSY-2 system, applying RMA to determine schedulability of real-time software designs and to recommend and verify (via RMA) design changes when needed. European Space Agency, On-Board Data Div., announces RMA will be the baseline methodology for its hard real-time operating sys. project. (The RMARTS Project had no direct contact with ESA.)

1990 - 2

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
			RMARTS Project continues to insure hardware support by seeing that RMA is used in revising FutureBus+ standard, which will be implemented by all major hardware vendors and is one of the Navy's Next Generation Computer Resource standards. RMARTS Project insures that standards for operating systems and programming languages allow the use of RMA by working with POSIX op. systems standard IEEE P1000.4, which results in inclusion of some basic aspects of RMA. Work to influence Ada 9X standard to encourage use of RMA algorithms is initiated. *Systems Design Manual* for IEEE FutureBus+ contains a chapter on how to use the FutureBus+ when designing real-time systems based on RMA. Research Institute for Computing and Info. Systems (RICIS) adapts instructional materials for use in training at NASA and its contractors. RMA theory referenced in part in real-time computing courses at Univ. of Mass., U. of Illinois, etc.	Introductory article on RMA and Ada published in *IEEE Computer*, 4/90. Two papers, "Maintaining Global Time in the IEEE FutureBus+" and "Real-Time Scheduling Support in FutureBus+", are accepted for publication by the *Real-Time Systems Journal* and *IEEE Micro*, respectively. These papers address design issues concerning support for real-time system development based on RMS theory and the IEEE FutureBus+. Instructional materials upgraded and incorporated into workshop at TriAda90 and the IEEE Real-Time Symposium.

1991

(Project was known as Rate Monotonic Analysis for Real-Time Systems)

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Work with the CMU Adv. Real-Time Tech. Project to extend the current analytical methods for assessing schedulability continues. A paper, "Fixed Priority Scheduling of Periodic Tasks with Varying Executive Priorities," coauthored by RMARTS, is accepted for presentation at the IEEE Real-Time Systems Symposium December 1991. Work begins to extend RMA to networks where distributed stations have to cooperatively schedule the network with incomplete information. (This work to be done by a CMU PhD candidate under the supervision of an RMARTS Project member.)	Technical note prepared describing several algorithms for assessing the schedulability of a task set with arbitrary deadlines. Direct support to BSY-2 provided: RMA data sheets developed and used to collect data for rate monotonic analysis of both individual and integrated CSCIs; rate monotonic analysis of individual CSCIs; tutorials presented to BSY-2 subcontractors; and a report on how to perform RMA on BSY-2 software designs. Work with NGCR Program to encourage development of a LAN that supports RMA principles. End-to-end scheduling issues are examined and IEEE 802.6 network standard is reviewed for applicability to the real-time domain.	Transition to research community for extension of RMA includes courses and/or research at Texas A&M in support of Space Station Freedom, U. Mass, Florida State, and the Univ. of Colorado. Nanotek announces single board computer products for FutureBus+, advertising support for RMA. The interface between FutureBus+ and other LAN standards is examined. Continuing support of the transition of FutureBus+ to the real-time community, including review of TI chipset designs, work with IBM on the real-time chapter of the *IEEE FutureBus+ System Configuration Manual*. Work with the real-time POSIX working group continues, including incorporation of proposals regarding priority inheritance protocols and processor allocation scope for threads that were balloted.	An overview of this chapter, "Real-Time Application Using IEEE FutureBus+," is accepted by *IEEE Micro* for publication. Prime contractors for PAVE PACE, a program developing the next generation architecture for AF avionics systems, are encouraged to use RMA. Boeing plans to incorporate RMA. A NASA report that discussed the results of a rate monotonic analysis of the Space Station Data Mgmt System software is reviewed. NASA Space Station studies data sheets developed for BSY-2.

1991 - 2

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
		An interim solution for users who want to begin using the sporadic server is developed, which is a modification of the sporadic server algorithm that can be implemented as an application-level Ada task.	Ada usage/performance specs are developed to ensure programmers following certain restrictions get the benefit of performance improvements in generated code and runtime system behavior. An MOU between SEI and System Designers, a compiler vendor, is signed as the basis of collaboration on this work. Similar MOUs with other compiler vendors are pursued. Plans are formed to develop an engineering handbook for the application of rate monotonic analysis to real-time systems. An intermediate step results in a draft report circulated for external review, "Rate Monotonic Analysis Adoption Rationale," which answers technical and managerial questions about RMA adoption. Sample problems and a taxonomy have been developed and are being reviewed.	NASA downsizing reduces extent of use of RMA. Project members convince vendors of operating systems and compilers to provide sporadic server support at their own expense to compensate. NASA and primes agree to use the additional options. Later NASA decides to use RMA for all space station software when applicable. RMA has been specified for use with space station on-board software as the means for scheduling multiple independent task execution. RMA will actually be built into the on-board operating system, and is directly supported by the ADA compiler in use. NASA Langley schedules RMA training. RMA is used in the nuclear partition of BSY-2 with RMARTS support. General Dynamics is applying RMA during development.

1991 - 3

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
			Data sheet concept, developed for BSY-2, will be incorporated in the handbook. Proposals for Ada 9X revisions are reviewed, and recommendations made. Work continues to encourage adoption of the sporadic server by Ada vendors and potential users. (The sporadic server is the scheduling mechanism introduced by RMA to provide enhanced schedulability and analyzability of aperiodic tasks.) Students in the SEI MSE program will develop a tool to support RMA. A commercially available tool set is found that supports RMA at a limited level. Work has been initiated to contact other CASE tool vendors about support for RMA. Vendors and contractors in attendance at TriAda91 and the IEEE Real-Time Systems Symposium indicate RMA adoption and extension through its inclusion in their products, training, and by descriptions in paper presentations.	Magnavox has applied RMA and documented its experience in a paper reviewed by RMARTS. Hughes applies RMA to the design of a radar warning receiver. Boeing Aerospace visits SEI to determine schedulability of software redesign of dual-processor SRAM-II using RMA. Raytheon's plan for the Patriot missile ground control software upgrade is reviewed with respect to its use of Ada and the applicability of RMA. A Raytheon employee who is a student in the SEI MSE program will analyze this software using RMA as part of his independent study. Software performance for an advanced radar system at Atlantic Aerospace is reviewed by RMARTS. A brief tutorial is also provided.

1992

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
	Advanced Real-Time Technology Project at CMU documents extensions to current analytical methods for assessing schedulability in a paper submitted to a special Real-Time Systems issue of *IEEE Transactions on Software Engineering*.	Data sheets (forms) for extracting processor utilization and memory requirements are developed and tested at GE Syracuse on nuclear partition CSCIs.	Prototype of RMA engineering handbook created to codify principles of RMA for software engineers. Work with Navy NGCR to encourage development of a LAN to support RMA principles. Project members use TaskGen tool to assess schedulability characteristics of runtime systems. SEI signs agreements with Telos and Tri-Pacific, who develop their own RMA courses for public consumption. Work with POSIX .12 to develop a LAN interface standard that meets needs of real-time applications. MSE students continue to work on a Real-Time Analysis Tool that performs schedulability assessments for software designs. Prototype handbook tested at Naval Air Warfare Station (NAWS) and Magnavox, and reviewed by external reviewers.	MSE student applies RMA to Patriot missile ground control software. Half-day RMA management tutorial presented at Washington Ada Symposium (WAdaS). Presentation on RMA management practices given at Software Technology Support Center annual conference (major military software conference). Telos and Tri-Pacific have vendor booths at the 1992 SEI Symposium to advertise their training courses. Courses listed in SEI product portfolio.

1992 - 2

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
			Prototype handbook (v2) restructured based on test results to allow contributions from external authors, and distributed to potential contributors. Prototype handbook (beta version) extended to include realistic case studies, and distributed to 350 reviewers. Work with Naval Warfare Center at China Lake to revise RMA tutorial to create baseline version for presentation at WAdaS. Work with Prof. Ruth Ravenel (U. Colorado) to develop a videotape-based short course, including exercises and course notes, on RMA. SEI and SPC develop memorandum of understanding so SPC can incorporate RMA into ADARTS and SPC real-time software development methodology. First annual RMA Users Forum (meeting of vendors and users) held in conjunction with SEI Symposium. SEI works with U. Virginia to support the prototyping effort for the Ada binding to the NGCR lightweight services intended for use in the real-time domain.	

1993

Promising Technology Selected	Key Limitations Addressed	Value and Transitionability Demonstrated	Self-Sustaining Transition	Widespread Use
(Is it potentially relevant?)	(Is it potentially practical?)	(Is it easily learned and/or built?)	(What support is needed for sustained usage?)	(Can it become common practice?)
			RMA handbook writing completed. Potential RMA handbook publishers contacted. Kluwer Academic Publishers selected to publish handbook. Negotiations continue with SPC on how to combine RMA with SPC ADARTS methodology. Telos sends consultant for training at SEI. 2nd RMA Users Forum planned. RMA handbook released. SEI works with Kluwer to consider a hypertext version of the RMA handbook.	RMA Video for SEI Technology Series taped at U. of Colorado by Prof. Ruth Ravenel. RMA column appears in *IEEE Computer* "Hot Topics" column. RMA "glossy" prepared. Telos offers RMA public courses. Tri-Pacific offers RMA tutorial at Ada-Europe. Consumer real-time market targeted for RMA with plans to attend and participate in 1993 and 1994 Embedded Systems Conference.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-93-TR-29	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESC-TR-93-203
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003
8c. ADDRESS (city, state, and zip code)) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO: 63756E; PROJECT NO.: N/A; TASK NO: N/A; WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) Technology Transition Push: A Case Study of Rate Monotonic Analysis (Part 1)	
12. PERSONAL AUTHOR(S) Prscilla Fowler and Linda Levine	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM TO	14. DATE OF REPORT (year, month, day) December 1993 15. PAGE COUNT 52 pp.
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (continue on reverse of necessary and identify by block number) adoption diffusion of innovation rate monotonic analysis (RMA) software technology transition technology maturation

19. ABSTRACT (continue on reverse if necessary and identify by block number)

This case study reports on efforts to transform rate monotonic scheduling theory from an academic theory into a practical analytical technique and to transition that technique into routine practice among developers and maintainers of software embedded in real-time systems.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF	22b. TELEPHONE NUMBER (include area code) (412) 268-7631 22c. OFFICE SYMBOL ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19