INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.B. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPRISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: W. Rance Cleaveland

Gender: ☒ Male ☐ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race: (Select one or more)
- ☐ American Indian or Alaska Native
- ☐ Asian
- ☐ Black or African American
- ☐ Native Hawaiian or Other Pacific Islander
- ☒ White

Disability Status: (Select one or more)
- ☐ Hearing Impairment
- ☐ Visual Impairment
- ☐ Mobility/Orthopedic Impairment
- ☐ Other _____
- ☒ None

Citizenship: (Choose one) ☒ U.S. Citizen ☐ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☐

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☒

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.

Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important tasks, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and is not a precondition of award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represendted groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

DEVIATION AUTHORIZATION (if Applicable)

DEVIATION AUTHORIZATION:

Deadline extended to 9/24/99 due to Hurricane Floyd by:
Frank D. Anger, Program Director
Software Engineering & Languages PHONE: 703-306-1911
NSF/CISE/C-CR

COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 99-2

PD98-2880 **09/24/99**

FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.)

SOFTWARE ENGINEERING AND LANGU

FOR NSF USE ONLY

NSF PROPOSAL NUMBER

9988489

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
				020657151	

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)

SHOW PREVIOUS AWARD NO. IF THIS IS
☐ A RENEWAL
☐ AN ACCOMPLISHMENT-BASED RENEWAL

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYMS(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE

SUNY at Stony Brook

AWARDEE ORGANIZATION CODE (IF KNOWN)

0028381000

ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE

SUNY at Stony Brook
Office of Sponsored Programs
Stony Brook, NY. 117943362

NAME OF PERFORMING ORGANIZATION, IF DIFFERENT FROM ABOVE

PERFORMING ORGANIZATION CODE (IF KNOWN)

ADDRESS OF PERFORMING ORGANIZATION, IF DIFFERENT, INCLUDING 9 DIGIT ZIP CODE

IS AWARDEE ORGANIZATION (Check All That Apply)
(See GPG II.D.1 For Definitions) ☐ FOR-PROFIT ORGANIZATION ☐ SMALL BUSINESS ☐ MINORITY BUSINESS ☐ WOMAN-OWNED BUSINESS

TITLE OF PROPOSED PROJECT **Heterogeneous Specification Formalisms for Reactive Systems**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PREPROPOSAL NO., IF APPLICABLE
$ **398,705**	**36** months	**05/01/00**	

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

☐ BEGINNING INVESTIGATOR (GPG 1.A.3)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.D.1)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG II.D.10)
☐ NATIONAL ENVIRONMENTAL POLICY ACT (GPG II.D.10)
☐ HISTORIC PLACES (GPG II.D.10)
☐ SMALL GRANT FOR EXPLOR. RESEARCH (SGER) (GPG II.D.12)
☐ GROUP PROPOSAL (GPG II.D.12)
☐ VERTEBRATE ANIMALS (GPG II.D.12) IACUC App. Date _____
☐ HUMAN SUBJECTS (GPG II.D.12)
Exemption Subsection _____ or IRB App. Date _____
☐ INTERNATIONAL COOPERATIVE ACTIVITIES: COUNTRY/COUNTRIES _____
☐ FACILITATION FOR SCIENTISTS/ENGINEERS WITH DISABILITIES (GPG V.G.)
☐ RESEARCH OPPORTUNITY AWARD (GPG V.H)

PI/PD DEPARTMENT
Department of Computer Science

PI/PD FAX NUMBER
516-632-8334

PI/PD POSTAL ADDRESS
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 117944400
United States

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Electronic Mail Address
PI/PD NAME **W. Rance Cleaveland**	**Ph.D.**	**1987**	**516-632-8448**	**rance@cs.sunysb.edu**
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

CERTIFICATION PAGE

Certification for Principal Investigators and Co-Principal Investigators:

I certify to the best of my knowledge that:

(1) the statements herein (excluding scientific hypotheses and scientific opinions) are true and complete, and
(2) the text and graphics herein as well as any accompanying publications or other documents, unless otherwise indicated, are the original work of the signatories or individuals working under their supervision. I agree to accept responsibility for the scientific conduct of the project and to provide the required progress reports if an award is made as a result of this application.

I understand that the willful provision of false information or concealing a material fact in this proposal or any other communication submitted to NSF is a criminal offense (U.S.Code, Title 18, Section 1001).

Name (Typed)	Signature	Social Security No.*	Date
PI/PD **W. Rance Cleaveland**			
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			
Co-PI/PD			

SSNs are confidential and are not displayed *ON FASTLANE SUBMISSIONS*

Certification for Authorized Organizational Representative or Individual Applicant:

By signing and submitting this proposal, the individual applicant or the authorized official of the applicant institution is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding Federal debt status, debarment and suspension, drug-free workplace, and lobbying activities (see below), as set forth in Grant Proposal Guide (GPG), NSF 99-2. Willful provision of false information in this application and its supporting documents or in reports required under an ensuring award is a criminal offense (U. S. Code, Title 18, Section 1001).

In addition, if the applicant institution employs more than fifty persons, the authorized official of the applicant institution is certifying that the institution has implemented a written and enforced conflict of interest policy that is consistent with the provisions of Grant Policy Manual Section 510; that to the best of his/her knowledge, all financial disclosures required by that conflict of interest policy have been made; and that all identified conflicts of interest will have been satisfactorily managed, reduced or eliminated prior to the institution's expenditure of any funds under the award, in accordance with the institution's conflict of interest policy. Conflict which cannot be satisfactorily managed, reduced or eliminated must be disclosed to NSF.

Debt and Debarment Certifications

(If answer "yes" to either, please provide explanation.)

Is the organization delinquent on any Federal debt? Yes ☐ No ☒

Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency? Yes ☐ No ☒

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:

(1) No federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.

(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, ''Disclosure Form to Report Lobbying,'' in accordance with its instructions.

(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME/TITLE (TYPED) **katherine L. MacCormack**		**09/24/99**

TELEPHONE NUMBER	ELECTRONIC MAIL ADDRESS	FAX NUMBER
516-632-4402	**kmaccormack@notes.cc.sunysb.edu**	**516-632-6963**

*SUBMISSION OF SOCIAL SECURITY NUMBERS IS VOLUNTARY AND WILL NOT AFFECT THE ORGANIZATION'S ELIGIBILITY FOR AN AWARD. HOWEVER, THEY ARE AN INTEGRAL PART OF THE INFORMATION SYSTEM AND ASSIST IN PROCESSING THE PROPOSAL. SSN SOLICITED UNDER NSF ACT OF 1950, AS AMENDED.

Heterogeneous Specification Formalisms for Reactive Systems

Rance Cleaveland* Gerald Luettgen†

September 22, 1999

Project Summary

The proposed research will be devoted to the development of novel techniques and tool support for *heterogeneous formal specifications* of reactive systems such as communications protocols, avionics systems and embedded software. Traditional formal-methods research in this area has focused on homogeneous approaches in which all analysis occurs within one framework. In practice, however, heterogeneous methodologies supporting multi-paradigm specifications prove very useful, for several reasons: different system components might be more naturally expressed in one specification formalism than another; different system components might be designed by teams favoring different specification styles; the same system might be specified in different notations at different stages in its design. To enable the rigorous analysis of heterogeneous specifications, the project will focus on the following lines of inquiry:

1. the development of a theory for the uniform treatment of operational and assertional system specifications, together with notions of specification refinement;
2. the study of mechanisms, based on those found in existing design and requirements languages, for composing specifications given in disparate formalisms into single specifications;
3. the implementation of automated tool support for verifying that one heterogeneous specification refines another;
4. the investigation of case studies involving avionics systems and communication protocols in order to assess the utility of the work.

This research will greatly enhance the benefits of formal system specification and verification by supporting the "interoperation" of different specification and verification technologies. It will also provide a sound semantic basis for analyzing the kinds of multi-paradigm behavioral specifications definable in design notations like the Unified Modeling Language (UML).

*Contact information: Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794-4400. Tel.: (516) 632-8448 (voice), (516) 632-8334, fax. E-mail: rance@cs.sunysb.edu. URL: www.cs.sunysb.edu/~rance/.

†Contact information: Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681-2199. Tel.: (757) 864-8003 (voice), (757) 864-6134 (fax). E-mail: luettgen@icase.edu. URL: www.icase.edu/~luettgen/.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.C.

Section		Total No. of Pages in Section	Page No.* (Optional)*
Cover Sheet (NSF Form 1207 - Submit Page 2 with original proposal only)			
A	Project Summary (not to exceed 1 page)	1	
B	Table of Contents (NSF Form 1359)	1	
C	Project Description (including Results from Prior NSF Support) (not to exceed 15 pages) **(Exceed only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**	15	
D	References Cited	9	
E	Biographical Sketches (Not to exceed 2 pages each)	4	
F	Budget (NSF Form 1030, including up to 3 pages of budget justification)	15	
G	Current and Pending Support (NSF Form 1239)	2	
H	Facilities, Equipment and Other Resources (NSF Form 1363)	1	
I	Special Information/Supplementary Documentation		
J	Appendix (List below.) **(Include only if allowed by a specific program announcement/ solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**		
	Appendix Items:		

*Proposers may select any numbering mechanism for the proposal, however, the entire proposal must be paginated. Complete both columns only if the proposal is numbered consecutively.

C Project Description

Reactive systems maintain ongoing interactions with their environments in order to control or affect them in some manner; examples of such systems include communications and electronic-commerce protocols as well as the programmable controllers found in applications such as heart pacemakers [105], automobile powertrains [107], active buildings [58], and avionics components [89]. As these examples suggest, reactive systems often function in safety- and business-critical applications and must therefore satisfy stringent dependability requirements. Despite their growing prominence, however, reactive systems remain difficult and expensive to design, debug and maintain. Two chief conceptual difficulties frequently confound designers: the *concurrency* in such systems, and their propensity for *nondeterminism*. In general, reactive systems include a number of simultaneously active subsystems, or *processes*, that can interact in subtle and often unanticipated ways. Nondeterminism arises from concurrency and from the unpredictability of a system's environment, and it complicates the task of diagnosing undesired system behavior by making erroneous executions difficult to reproduce. Existing development practice relies on the use of intensive testing; while hard numbers are difficult to obtain for reactive systems, a wealth of anecdotal evidence suggests that testing budgets for reactive systems can represent 50% or more of development costs. Nevertheless, system errors still arise, with sometimes devastating consequences [88].

Formal specification and verification offers an appealing approach for coping with the problem of designing dependable reactive systems. Formal specification refers to the development of mathematically precise models of systems and their requirements; verification is then the process of determining whether a system description satisfies its requirements. Although relatively unused by the software engineering community, mathematical modeling constitutes a cornerstone of design in other engineering disciplines. In fluid dynamics, for example, the analysis of mathematical models of air frames enables engineers to analyze quickly the ramifications of different design decisions. Although such analyses do not obviate the need for testing the real product, they greatly reduce the amount of (expensive) time spent studying physical design artifacts in wind tunnels. In the same manner, specification and verification methodologies can reveal errors, omissions and inconsistencies in designs of reactive systems. Research in this area has already led to the emergence of fully automated verification techniques, such as *temporal-logic model checking* [28], which have begun to make formal methods more accessible to nonspecialists. Industrial interest has also grown, especially in the hardware community, and several vendors, including Cadence, Synopsys, and i-Logix, now sell tools that include support for formal specification and verification.

Heterogeneous specifications. Despite the potential benefits of formal specification and verification in reactive-system development, these techniques are rarely employed in current design practice. One factor impeding the uptake even of automated formal methods is that relatively little attention has been paid to issues of "interoperability". Specifically, while a number of useful specification and verification methodologies for reactive systems have been developed and successfully applied, little work has been done on ways of combining different specifications into mathematically intelligible composite specifications. We refer to such multi-paradigm specifications as *heterogeneous specifications* to distinguish them from more traditional *homogeneous* ones. Heterogeneous specifications have strong practical motivations:

- *Different specification formalisms are tuned for different applications.* To specify containing a mixture of hardware, software, and off-the-shelf components, a design team might use oper-

ational formalisms (e.g. process algebra [7, 16, 73, 104] or Statecharts [69]) for the hardware and software and an assertional notation (e.g. temporal logic [59]) for describing properties of the others. Both types of specification need to interact in order for the behavior of the full system to be analyzed.

- *Specifications come from different sources.* When system components are developed by different teams they may be specified in different formalisms, depending on the characteristics of the component under consideration and on the background of the engineers responsible for it. Developing analyzable models for systems assembled from such components would give engineers earlier feedback on their design decisions than is currently possible.

- *Specification concerns change in the course of the design cycle.* System requirements are often stated using assertions, whereas system designs must be operationally concrete. A unified formalism would allow users to refine parts of their specification into detailed, operationally oriented designs while leaving other parts abstract, and then analyze the behavior of the resulting "mixed" specification.

The appeal of heterogeneity in (informal) specifications has already been recognized within the software-design community, as evidenced by the interest in the *Unified Modeling Language* (UML) [17, 78, 79, 117] for object-oriented and reactive systems [54]. UML includes variants of *Statecharts* [69, 71] and *Message Sequence Charts* [77] as notations for modeling system behavior; it also provide a rudimentary notation, the *Object Constraint Language* (OCL) [127, 128], for defining assertional constraints on system activity. However, while the syntax of these UML features is described relatively precisely, the language as yet has no semantics for specifications containing both operational and assertional information; even the individual sublanguages of UML remain subjects of active study from a semantic standpoint [70, 91, 93, 95, 101, 123, 124]. Thus UML tools are not currently able to provide support for rigorously analyzing the dynamic behavior of system specifications.

Research agenda. The goal of this research project, which involves a collaboration between researchers at SUNY at Stony Brook and the Institute for Computer Applications in Science and Engineering (ICASE)[1], is to develop mathematically well-founded, practically useful techniques for constructing and analyzing heterogeneous specifications of reactive systems. The specific topics to be investigated include the following.

A uniform theory for operational and assertional specifications. We propose to develop a semantic framework based on on Büchi automata [122] for representing heterogeneous specifications containing operational and assertional content. The theory will be equipped with a refinement ordering that extends traditional testing preorders [72] and that is compatible with the satisfaction relation of linear-time logics [114].

Mechanisms for composing heterogeneous specifications. We plan to investigate combinators supporting the development of structured heterogeneous specifications, and to define design notations based on these operators. We will devote special attention to languages mixing

[1]ICASE is a private non-profit research institute run by the University Space Research Association (USRA), which in term is under the auspices of the National Academy of Science. ICASE is located at the NASA Langley Research Center in Hampton, Virginia. Despite its location, ICASE is *not* a Federal agency. More information about ICASE and its mission may be found at `www.icase.edu`; USRA's web site is `www.usra.edu`.

Statecharts [69], which can be used to describe operational behavior; Linear-time Temporal Logic (LTL), which supports the description of assertional constraints; and Message Sequence Charts (MSCs) [77], which provides a means for specifying systems in terms of scenarios.

Tool support for analyzing heterogeneous specifications. We will implement our heterogeneous specification formalisms in an automated verification tool, the Concurrency Workbench [44, 46]. In particular, we will develop algorithmic support for computing the refinement preorder defined as part of this research effort, and we will build appropriate front-ends for the tool so that users may build and analyze heterogeneous specifications.

Case studies. In order to evaluate our work, we will conduct two case studies that together will exercise all aspects of our research. The first case study is concerned with the design of future flight guidance systems, while the second deals with the specification and analysis of a safety-critical, fault-tolerant communications bus used in aeronautical applications.

The outcome of this research will be a collection of tools and techniques permitting the construction and formal analysis of heterogeneous reactive-system specifications. Using these results, engineers will be able to (1) specify reactive-system components and their requirements using notations best suited for the particular applications, (2) share specifications and verifications across different projects and organizations in a mathematically robust way, and (3) reuse previously-developed specifications.

The remainder of this project description is structured as follows. The next section summarizes relevant background material, while the one following provides more details on our research plans.

C.1 Background

This section reviews some of the terminology and concepts that will be used in the remainder of the proposal.

Process Algebra. *Process algebras* [7, 72, 73, 104] are theories for modeling and reasoning about reactive systems. A number of different process algebras have been developed—CCS [103, 104] being perhaps the best-known—but all share the following key ingredients.

- *Compositional modeling.* Process algebras provide a small number of constructs for building larger systems up from smaller ones. CCS, for example, contains six operators in total, including ones for composing systems in parallel and others for choice and scoping.

- *Operational semantics.* Process algebras are typically equipped with a semantics that describes the single-step execution capabilities of systems. Using this semantics, systems represented as terms in the algebra may be "compiled" into labeled transition systems ("state machines").

- *Behavioral reasoning via refinement.* Process algebras provide behavioral relations as a means for determining when one system "refines" another. These relations may be equivalences, which stipulate that one system correctly implements another if they "behave the same", or preorders, which order processes on the basis of the "quality" of their behavior. Often these relations are *congruences*; this means that related systems may be substituted for one another inside larger contexts in a relation-preserving manner.

In a process-algebraic approach to system verification, one typically specifies a system by defining another system describing the desired *high-level* behavior. One then establishes the correctness of a design or implementation with respect to such a specification by showing that it behaves the "same as" the specification (if one is using an equivalence) or by showing that it behaves "better than" the specification (if one is using a preorder). The advantages to an algebraic approach are the following.

- *System designers need learn only one language* for specifications and designs.
- *Related processes may be substituted for one another* inside other processes. This makes process algebras particularly suitable for the modular analysis of layered, hierarchical designs, since specifications and correct designs may be used interchangeably inside larger systems.
- *Processes may be minimized* with respect to equivalence relations before being analyzed; this sometimes leads to orders of magnitude improvement in the performance of verification routines [58].

Process algebras have been studied extensively since the late 1970's, mostly in Europe; notable examples besides CCS include CSP [73], ACP [9], and LOTOS [16].

Temporal Logics. *Temporal logics* [59, 96, 121] support the formulation of assertions about a system's behavior as it evolves over time. Using temporal logic, one can specify a system by providing a collection of formulas that the system is supposed to satisfy. Typically, such a collection would include a list of *safety properties* defining what should always be true of a system and a set of *liveness properties* describing conditions that a system must eventually satisfy. As an example, one might require that a communications protocol always be deadlock-free (a safety property) and that whenever it is given a message, it is guaranteed to deliver it eventually (a liveness property).

The advantage of temporal logic is that it allows designers to focus on constraining the aspects of system behavior they are interested in without requiring them to say anything about behavior they are not concerned about. Temporal logic has been the subject of intensive research over the past 20 years; numerous variants, including Computation Tree Logic (CTL) [28], CTL* [60], the modal/propositional mu-calculus [82, 121], and Linear-time Temporal Logic (LTL) [114], have been designed and techniques for proving systems correct using the temporal logic developed [26, 68, 97, 118]. When the systems in question are finite-state, these "proofs of correctness" may be conducted automatically using *model-checking* algorithms [5, 11, 13, 23, 27, 28, 50, 64, 65, 92, 115, 125]. The practical utility of model checking has been demonstrated on a number of case studies (see [30] for an overview) and has begun attracting industrial attention [66, 116]. Thorough surveys of temporal logic in Computer Science may be found in [59, 121].

The Concurrency Workbench and the Process Algebra Compiler. The Concurrency Workbench [42, 43, 44, 46] is an easily customized tool for verifying finite-state systems described in process algebras. The key feature of the system is its modular design and concomitant flexibility. The system is built around three generic algorithms: one for computing behavioral equivalences (the general equivalence is based on *bisimulation equivalence* [104]), one for computing preorders (the general preorder is based on the *divergence preorder* of [126]), and one for *model checking* in the modal mu-calculus [82, 121], an expressive temporal logic. The system uses the first two of these generic routines to compute a number of different equivalences and preorders by combining them with suitable *process transformations routines* [33, 34]; to decide a given relation, the Workbench

applies the appropriate transformation to the processes in question and then runs the implied generic routine on the transformed processes. This structure makes it easy to add new process relations to the Workbench—just determine the appropriate process transformation and apply the indicated general procedure.

The inclusion of these different verification techniques permits different styles of correctness checking to be carried out, and it facilitates the development of methodologies that employ more than one of these techniques [49, 58, 87, 126]. The software has been acquired by numerous sites around the world (18 countries on five continents, at last count) and has been used successfully on numerous case studies [14, 22, 25, 31, 39, 58, 112, 120].

The *Process Algebra Compiler* (PAC) [41, 47] is a front-end generator for the CWB. Given formal descriptions of a process algebra's syntax and operational semantics, the PAC generates code needed to enable the CWB to analyze designs given in the language. This makes it very easy to add new language support to the CWB, and to add new operators to existing languages. The PAC employs two key optimizations that make the generated front ends very efficient; indeed all six front-ends in the current release (1.11) of the CWB are PAC-generated.

C.2 Proposed Work

As concrete motivation for the proposed work, consider the following examples.

1. A system designer wishes to design a communications protocol intended to work over a commercial off-the-shelf Medium. S/he develops state-machine models of the Sender and Receiver entities and wants to ensure that a property ϕ, asserting that all messages sent are eventually received, is satisfied. However, all that is known about Medium are the properties informally described in the programmer's guide from the manufacturer.

2. In a variation on the first example, instead of being outsourced Medium is constructed by another team in the same organization, and the designer has access to that team's specification. S/he therefore checks whether Sender|Medium|Receiver satisfies ϕ, where | represents (asynchronous) parallel composition, using model checking. The check fails, owing to apparent faulty behavior in Medium. The team responsible for Medium asserts that in fact the behavior in question, while theoretically possible, never "really" happens.

3. A designer of an avionics subsystem has a temporal property γ that his/her system should satisfy, and s/he also has an architecture and component specifications inherited from a previous project. S/he would like to know whether modifying the existing specifications so that the final system satisfies property γ' will ensure that γ is also met.

All of these problems can be addressed using heterogeneous specifications. In the first example, the designer could proceed by formulating a property ϕ' representing his/her assumptions about Medium and then checking whether or not the heterogeneous specification Sender|ϕ'|Receiver satisfies ϕ. In the second, the designer can formulate a temporal property ϕ'' expressing a "fairness constraint" forbidding the offending execution and then check whether or not the heterogeneous specification Sender|(Medium$\wedge\phi''$)|Receiver satisfies ϕ. Here $\wedge$ represents *logical conjunction*. In the final example, the designer might take the partial design PD and check whether the heterogeneous specification PD$\wedge\gamma'$ satisfies γ. Note that in general, checking whether γ' logically implies γ will not suffice, since γ' might refer only to internal interactions among the components in the design and not mention the observable system properties referenced by γ.

Approaches to the abovementioned problems have been studied independently in the literature, although the scenarios just described do have twists that complicate the application of existing work. The first problem may be seen as an instance of the *modular verification* problem studied for example in [67], the twist being that the notion of parallel composition we use here is asynchronous instead of synchronous. The second problem involves the use of fairness constraints, which have been well-studied in the temporal-logic and model-checking literature [29, 62, 63, 96]. The twist here is that in contrast with existing work the fairness constraint is *localized* to a single subsystem; indeed, it is completely feasible using heterogeneous specifications to have different components governed by different fairness constraints. The final problem can be seen as an example of *assume/guarantee reasoning* [80, 86] or *modular model checking* [84]; the twist here is that the assumption applies not to a missing component but to missing functionality in existing components.

Existing work could be adapted to cope with the twists just described. However our point of view is that all three kinds of problems, and others, may be treated and solved in uniform way via heterogeneous specifications. The remainder of this section discusses our ideas in more detail. At the outset, however, we wish to clarify the setting in which we work.

- System specifications will be *event-* rather than state-based; our emphasis will be on system models and logics that focus on the events system engage in.
- The temporal logics we consider will be linear-time, and formulas will be interpreted with respect to sequences of *events*, rather than sequences of states, as is more traditional. A system correctly implements such a formula if every maximal sequence of events the system can engage in satisfies the formula.

We now turn to a description of our specific research agenda.

C.2.1 A Semantic Theory of Heterogeneous Specifications

In order to develop a mathematically coherent framework of heterogeneous specifications we must first define a common semantic formalism into which different specifications can be translated. We specifically plan to focus on a unifying formalism for process algebra and Linear-time Temporal Logic, or LTL [114], for the following reasons.

1. Both theories contain a sparse but expressive set of constructors and have a well-studied semantics.
2. Process algebras may be seen as paradigmatic design notations for reactive systems, and LTL is an exemplar of assertional notations for defining requirements of reactive systems.
3. The two theories support quite different specification and verification styles, with process algebra favoring refinement-based approaches and temporal logic supporting a scenario-oriented, "property-at-a-time" style. Unifying these theories will therefore provide insight into general issues in heterogeneity.

The specific issues to be confronted are described below.

Operational models of process algebra and LTL. We plan to develop *operational models* based on labeled transition systems and use these as a semantic foundation for heterogeneous specifications mixing process algebra and LTL. Labeled transition systems already form the basis for the semantics of process algebra as well as other design notations such as Statecharts [69] and SDL [76]. Such languages typically have an operational semantics describing the "atomic execution steps" system descriptions may engage in. Such a semantics may then be used as a basis for "compiling" designs into labeled transition systems that encode all the behavior the design is capable of.

Special kinds of labeled transition systems known as *Büchi automata* [122] may also be used to give a semantics to LTL formulas [125]. Büchi automata resemble traditional finite-state machines: they contain states and transitions, and certain states are designated as "accepting". However, Büchi automata are intended to accept infinite rather than finite sequences; such a sequence is in the language of a Büchi automaton if the machine goes through a accepting states infinitely often while processing the sequence. The connection with LTL is as follows: for any LTL formula one may build a Büchi automaton accepting exactly the sequences that satisfy the formula.

Based on these observations we will investigate suitably annotated labeled transition systems as a semantic basis for heterogeneous specifications. The following questions will need to be confronted.

1. How can different notions of choice be accommodated? In process algebra one encounters a distinction between *internal* and *external* choice, with the former being resolved within the system and the latter being resolved by the environment. In LTL no such distinction exists. However, the branching found in Büchi automata appears to be sensibly construed as internal choice. We will consequently endow our models with capabilities for expressing both kinds of choice using standard techniques from process algebra [104].

2. How will termination and divergence be catered for? Process algebras allow the modeling of systems that may terminate or deadlock as well as ones that can engage in livelocking. The latter phenomenon is often called *divergence* and plays a key role in certain refinement relations [19, 72]. LTL does not have these notions; moreover, the traditional semantics of LTL is only given with respect to infinite sequences, implying that the language is incapable of expressing constraints on "terminal" behavior. However, the semantics of LTL may be altered slightly to include finite sequences as well, and we will do so in order to allow LTL formulas to be interpreted with respect to "terminating" sequences as well as infinite ones. It also appears that divergence has a logical interpretation in terms of an LTL formula that is everywhere true, owing to the fact that refinement orderings such as the must-preorder [72] typically view all processes as correct implementations of divergent ones.

3. How can "eventuality" constraints be defined? Certain LTL formulas naturally impose eventuality constraints on their models, and these are reflected in the Büchi acceptance condition: for a Büchi automaton to accept a sequence, it is always the case that eventually an accepting state must be entered. Traditional process algebras have no similar notion. To cater for it, we will endow the labeled transition systems we study with accepting states in the Büchi style.

4. Can "pure process algebra" and "pure LTL" be embedded in the model? As a sanity check on our operational formalism we will identify the classes of submodels needed to embed "pure processes" (i.e. ones built solely from process algebra constructs) and "pure formulas." It appears that labeled transition systems in which every state is "accepting" will suffice for

embedding process algebra specifications, while models in which all choices are internal will have the expressiveness necessary to encode LTL.

We refer to the labeled transition systems we plan to develop as *Büchi labeled transition systems.*

Refinement and Büchi labeled transition systems. Having defined a class of operational models we next will investigate *behavioral preorders* on this set with a view toward capturing when one model refines/implements/implies another. We wish the relation to be compatible with existing notions of refinement in process algebra and with LTL satisfaction, in the following sense.

- If the models being compared are both "pure process" models in the sense described above, then one refines another in the heterogeneous setting exactly when it does so in the traditional homogeneous setting.
- If one model corresponds to an LTL formula while the other is a "pure process", then the latter should refine the former exactly when it satisfies the corresponding LTL formula in the traditional sense.

To achieve both of these goals we propose to investigate extensions to the traditional *testing* theories of DeNicola and Hennessy [53, 72]. The original work was done in the setting of process algebras and consists of the following elements.

- *Tests as transition systems.* Tests are labeled transition systems with certain states designated as "successful".
- *Test application via parallel execution.* To apply a test to a system, one runs the two in parallel. An execution of the parallel combination is *successful* if the test enters a successful state.
- *Orderings based on outcomes.* Because of nondeterminism one may distinguish between two kinds of success: *may* and *must.* A process may-passes a test if, when the test is applied, at least one execution is successful; it must-passes the test if every execution is successful. Using these ideas one may define the obvious orderings on systems: one is may-less than another if every test the lesser one may-passes is also may-passed by the greater one, and similarly for must.
- *Alternative characterizations.* To simplify reasoning about the may and must orderings, alternative characterizations are developed that appeal only to information regarding system traces and *acceptance sets* [72]. These alternative characterizations also provide a basis for computing the orderings over finite-state systems.

One appealing aspect of the must-preorder is that for process algebras such as CCS and CSP, it coincides with the *maximal trace congruence.* That is, it is the coarsest relation that relates systems on the basis of their maximal traces (using reverse subset containment) while still allowing "greater" systems to be freely substituted for "lesser" ones inside system descriptions. This coincidence is suggestive for our purposes, since LTL formulas are satisfied by systems when the latter's "maximal traces" are contained in the sequences satisfying the former.

To define a testing-based theory of refinement we must address the following technical questions.

1. How do we define testing? We expect to adopt the DeNicola/Hennessy approach outlined above: tests will be Büchi labeled transition systems enriched with special success states. Defining an appropriate notion of test application will likely be subtle, owing to the presence of Büchi accepting states in both systems and tests.

2. What are the alternative characterizations of the must and may preorders? To simplify reasoning about our relations, we propose to investigate alternative characterizations. We expect that Büchi language containment will feature in these in one form or another.

3. Are the two criteria defined at the beginning of the previous paragraph satisfied? We have some reason to believe they might be, given the correspondence between traditional must-testing and maximal-trace inclusion.

Computing refinement. When two Büchi labeled transition systems are finite-state, we expect that the must-ordering just defined will be computable, and we propose to investigate algorithms for calculating it (i.e. for determining whether or not two finite-state systems are related). We anticipate that existing approaches [34] for the traditional preorder will be applicable; the challenge will lie in correctly accounting for Büchi acceptance information. Existing work on model checking using Büchi automata [51, 125] may also prove useful.

C.2.2 Compositionality and Heterogeneous Specifications

After developing a mathematical foundation for heterogeneous specifications, we next propose to investigate mechanisms for assembling larger specifications out of smaller ones. Specifically, we plan to study formalisms that combine *process algebras*, *Linear-time Temporal Logic* (LTL), and *Message Sequence Charts* [77], a graphical scenario-oriented notation for specifying systems. The following describes our ideas in more detail.

Combining CCS and LTL. To begin our study of linguistic support for heterogeneous specifications we propose to investigate a language, which we refer to as *CCS+LTL*, that combines the operators of the process algebra CCS [104] and LTL [114]. Such a language would allow the development of specifications that freely intermix the design-oriented concepts of CCS with the requirements-centered ones of LTL; in particular, one could conjoin a system description with a formula defining a fairness constraint, or put a formula in parallel with a process description. Both languages contain a small number of well-defined operators, and this lack of syntactic clutter will allow us to focus on general semantic issues involved in intermingling operational and assertional formalisms. The specific technical questions to be addressed include the following.

1. How should the operational semantics for CCS+LTL be defined? Existing definitions of operational behavior for the two formalisms differ substantially. The semantics of CCS is given in the Structural Operational Semantics (SOS) style, with rules defining how transitions of a system can be inferred from transitions of its subsystems. Büchi-automaton constructions for LTL formulas on the other hand favor tableau-based approaches [64, 125] that are not obviously compositional in the structure of formulas. Nevertheless, several of the logical operators appear to have analogs as process combinators (conjunction resembles the *synchronous* parallel composition operator of CSP [20], for example), and our early investigations suggest that LTL may be given an SOS also, with special rules for defining Büchi accepting states. If

this indeed the case, then we will study how similar rules for accepting states may be defined for the process operators as well. Our anticipated result will be a semantics in the form of SOS rules defining the single-step transitions and Büchi acceptance status of heterogeneous specifications given in CCS+LTL. These rules would then form the basis of a procedure for generating Büchi labeled transition systems from CCS+LTL specifications.

2. Which operators preserve refinement? As Section C.1 indicates, a hallmark of process algebra is compositional reasoning using (pre)congruences (i.e. substitutive behavior equivalences and refinement orderings). We wish to investigate which of the operators in CCS+LTL respect the must-ordering defined previously. Our intuitions suggest that all of them will, except for the CCS choice operator (a problematic operator for other theories also [53, 104]), but this requires further study. The alternative characterizations of the refinement ordering should prove valuable in this undertaking.

Statecharts and LTL. We also propose to develop a heterogeneous formalism that intertwines LTL with Statecharts (more specifically, with the structure-respecting fragment of Statecharts presented in [95]). This line of research is driven by practical concerns: one of the systems we propose to study later in the proposal (see Section C.2.4) makes heavy use of Statecharts, and the notation is generally beginning to attract significant attention from engineers of reactive systems.

In previous work [93] we gave a process algebra for Statecharts, and we expect to use this work, in combination with the framework we will develop for merging CCS and LTL, to define a combined Statecharts+LTL formalism. The main challenge to be faced stems from the distinction between "micro-steps" and "macro-steps" in the semantics of Statecharts; the latter represent "real" computation steps and are defined in terms of the former. Jibing the semantics of LTL with this two-level view may necessitate slight alterations to the interpretation of LTL formulas.

Message Sequence Charts. A chief virtue of temporal logics such as LTL is that they allow a form of "scenario-based" specification; users can define desired aspects of system behavior independently of one another. However, LTL often proves difficult to use in practice because the operators provided by the logic often do not map easily onto the operational intuitions of system designers. *Message Sequence Charts* [77] (MSCs) represent a widely-used alternative notation for describing "scenario-based" system requirements. Originally conceived of as a graphical means for describing the flow of messages between entities in a distributed system, MSCs also include structuring mechanisms allowing nondeterministic choice and iteration to be captured [77]. Analysis tools [3, 8, 113] and model-checking algorithms [4] have also been developed for such "structured" MSCs. Other enrichments to the MSC formalism have been studied in recent years, including notions of liveness [52, 85] and of MSCs as *proscriptive* requirements (i.e. reflecting a complete account of allowed system behavior) as well as prescriptive ones (i.e. describing a subset of required behaviors) [18, 21]. These latter extensions bring MSCs closer to temporal logic in terms of expressiveness; indeed the commercial tools SDT (`www.telelogics.se`) and CS Verilog (`www.verilogusa.com`) include facilities for treating MSCs as requirements specifications.

We wish to investigate the development of heterogeneous specification formalisms combining process notations such as CCS and Statecharts with MSCs. In this endeavor we will focus on the behavioral aspects of MSCs rather than the architectural information they also contain; in particular, we hope to be able to give a semantics to MSCs extended with liveness [52] in terms of the Büchi labeled transition systems described earlier in this proposal. For reasons of technical

convenience we propose to base our work on the process-algebraic treatment of MSCs given by Reniers and Mauw [99, 100]; in particular, we hope to adapt their SOS rules to our richer setting in which Büchi acceptance information must be included.

C.2.3 Tool Support for Heterogeneous Specifications

To test our ideas on meaningful case studies, and to make the theory accessible to nonspecialists, we propose to develop tool support in the context of the Concurrency Workbench [44, 46]. The specific tasks we wish to undertake include the following.

Implementation of support for CCS+LTL heterogeneous specifications. The Workbench already includes support for CCS and LTL; allowing heterogeneous specifications in CCS+LTL will therefore necessitate: altering the front end of the tool to accept heterogeneous specifications; modifying the internal Workbench data structures for labeled transition systems to accommodate Büchi information; implementing algorithms for computing the new refinement relation; and computing diagnostic information when the refinement relation fails to hold. We plan to use the Process Algebra Compiler (PAC) to produce the front-end; in addition to the syntactic routines for CCS+LTL, the PAC will also produce implementations of the necessary semantic routines, provided the semantics of CCS+LTL can be given as SOS rules.

Implementation of support for Statecharts+LTL+MSC specifications. We also plan to develop tool support for specifications of synchronous systems featuring combinations of LTL and textual versions of Statecharts [93] and Message Sequence Charts. Given the infrastructure implemented in the Workbench for handling CCS+LTL for the refinement relation described above, this effort will amount to the generation of a front-end for the combined textual language. We will again use the PAC in this effort.

A graphical front-end. The Statecharts+MSC front-end described above suffers from the fact that it forces users to develop their Statecharts and MSC specifications textually. We would like to investigate an integration of the Workbench analytical facilities with a graphical editor for Statecharts and MSCs so that users may enter their designs using the traditional graphical syntax of these notations.

C.2.4 Case Studies

To evaluate our results, we plan to conduct two significant case studies of interest both to NASA and the aviation industry. Both concern existing and future avionics systems. Together, the case studies will exercise all aspects of our work, including the semantic theory for multi-paradigm specifications; the unified language combining Statecharts, Linear-time Temporal Logic, and Message Sequence Charts; and the automated tool support. We expect some of the results of these investigations to be of great interest to the avionics community, since the applications under study are already, or will soon be, in daily use. In the following, we briefly describe each case study, argue why their conduct requires heterogeneous specification techniques, and point out the particular challenges involved.

The design and analysis of flight guidance systems. *Flight guidance systems* (FGSs) are parts of airborne flight control systems. They continuously determine the difference between the actual state of an aircraft – its position, speed, and attitude as measured by its sensors – and its

desired state as indicated by the crew or flight management system. When a difference is detected, the FGS generates commands to minimize this difference, which the autopilot may then translate into movements of the aircraft's actuators. Current research in FGSs by Rockwell Collins and NASA includes their formal designs and analyses [24, 102].

As with other systems, the design of FGSs includes two major tasks, namely the specification of abstract system requirements and the development of the detailed system design. However, FGS design is simplified by the fact that the *system's architecture is known in advance*, since avionics engineers have decades of experiences with these systems. This allows engineers to annotate requirements of FGSs directly on the architectural components to which they belong. For this task, designers often use a simple, assertional language which allows them to specify functional and reactive system behavior via pre-/post-conditions and invariants. Usually, a different team of engineers is then responsible for deriving the detailed design of the FGS under consideration. This team's work starts with the annotated architecture and successively refines and transforms the assertions into operational content, thereby utilizing a mixed assertional/operational specification notation, until the final, detailed, and fully-operational design is arrived at. The final design can then be semi-automatically implemented into hardware and/or software. Also the formal analysis of components of FGSs is of utmost importance to the aviation industry in order to meet the high safety standards imposed on avionics components by the government. FGSs need to obey not only mandatory properties regarding their proper functional behavior, but also properties suggested by experts in human factors and human/machine interfaces for preventing pilots from getting confused about the actual state of the FGS. This kind of confusion, which is also referred as *mode confusion* [90], has recently been identified as a considerable source for avionics incidents and accidents [75].

Current specification technologies do not provide any semantic basis for the abovementioned kind of refinement, nor do they support the reasoning necessary for establishing that each refinement step preserves the initially imposed requirements. The proposed work is targeted towards enabling this facet of refinement-based system design. In particular, the proposed mixed design language is suited for specifying the architectural and operational design of FGS, which can be modeled by Statecharts, as well as for specifying the assertional requirements, which can be conveniently expressed in Linear-time Temporal Logic. In the literature, it has previously been shown that linear-time logics are well-suited for specifying both mandatory properties and mode-confusion properties of FGSs, and that automata-based model checking is able to efficiently carry out the analyses [94]. Since our proposed refinement mechanism is compatible with this approach to verification, our research results should provide a sound semantic basis for the desired design methodology for future-generation FGSs.

A bus architecture for integrated modular avionics. The second case study involves the formal specification and analysis of a *safety-critical, fault-tolerant bus* which was originally developed by Honeywell and is known as SAFEbus™ [74]. A variant of it has then been standardized by the Airlines Electronic Engineering Committee as ARINC 659 [1]. The bus is used in the Boeing 777 Airplane Information Management System as part of an architecture for Integrated Modular Avionics (IMA), as well as in several other avionics applications.

The SAFEbus is an intelligent backplane bus which may be thought of as a distributed kernel that guarantees certain fault-tolerant services and properties, such as *fail-silent bus interfaces* and *redundancy management*, for IMA. These capabilities are achieved by a combination of replicated hardware and *communication protocols* for enabling the exchange and synchronization of information. The design of the SAFEbus involves a collection of components on two major, tightly

integrated layers: the bus protocol layer and the redundancy management layer. Whereas the bus protocol layer is mostly operationally specified on a detailed level using *timing diagrams*, the redundancy management layer is mostly given on an abstract level using an assertional notation for encoding loosely coupled, causal constraints. Moreover, the applications utilizing the bus are treated as black boxes which are known to observe a few simple assumptions about their behavior. Hence, the information available regarding these applications is as sparse as the one for off-the-shelf components. Although the complete SAFEbus was specified by a single team, the different characteristics of different bus components forced engineers to employ multiple specification techniques.

The challenge of the formal specification and analysis of the SAFEbus architecture lies in the fact that many components, although having different characteristics and tasks, are highly integrated into one design. Thus, the layering and separation of concerns is not as simple for SAFEbus as for other fault-tolerant platforms. We suspect that most components concerned with redundancy management can be specified in temporal logics, whereas the timing diagrams for describing the message sequencing in communication protocols can be converted into Message Sequence Charts. The analysis of the bus involves the verification of safety and liveness properties which the integrated bus protocols need to obey in order for the redundancy management to function properly. The theory and tool support developed as part of this proposal will should enable us to perform these checks.

C.3 Results from Prior NSF Support

Award: CCR-9257963/CCR-9996312
Amount: $212,500
Duration: 9/15/92–8/31/99
Title: NSF National Young Investigator Award

This section reports on results obtained during the PI's most closely related NSF-funded project during the past five years. The PI has had several other projects with the NSF during this time frame; the relevant project numbers include CCR-9120995, INT-9247478, CCR-9257963, CCR-9505662, INT-9603441 and CCR-9804091/CCR-9996086. Some awards, the one under discussion included, have two numbers because the grants were transferred to SUNY at Stony Brook when the PI started his current position there in 1998.

Research in this project focused on the following areas: model-checking algorithms for finite-state systems; uses of abstraction in handling state explosion; approaches to modeling real-time and probabilistic systems; and tool development. The remainder of this section discusses each in turn and concludes by mentioning the impact of this project on educational and human resources.

Model checking. The work in model checking focused on the development of on-the-fly techniques for determining if systems satisfy formulas and on the investigation of the modal mu-calculus [82] as a practical basis for "generic" model checking.

The interest in on-the-fly techniques stems from the fact that they compute information needed to determine whether or not a system satisfies a formula in a *demand-driven* manner. Among other things, this allows the state space of a system to be constructed incrementally. Thus, if only a few states of a system need to be examined in order to ascertain the truth or falsity of a formula (as is typically the case early in the development of a system, when it is likely not to have the

properties a designer desires), then the model checker uses very little storage. Traditionally, on-the-fly algorithms have exhibited worse time complexity than so-called *global* approaches and hence have not been used extensively in existing tools. In this project we overcame this drawback by giving on-the-fly algorithms for two important logics, CTL* [13] and the L_2 fragment of the mu-calculus [11], whose performance matches that of the best global algorithms [61]. An implementation of the algorithm in [11] has been used successfully in several case studies [10, 14, 39] and is included in the current release of the Concurrency Workbench of North Carolina [46].

Regarding the use of the mu-calculus as a basis for efficient model checking, while it has long been known that the expressiveness of the mu-calculus exceeds that of all known temporal logics, these results have not had a practical impact on model-checking tools. The reason for this is that the known translation procedures for temporal logics such as CTL* require significantly more time than existing special-purpose model-checking algorithms for these logics. We showed that this inefficiency is not inherent by giving an efficient translation for CTL* and other temporal logics into a version mu-calculus in which one can write formulas as equations [12]. The combination of the complexities of the translation procedure and the model-checking procedure for the fragment of the mu-calculus needed for the translations turns out to match the complexity of the best existing procedures.

Abstractions. The PI's work on abstractions focused on techniques for limiting state-explosion due to *data profusion*. The main practical impediment to automatic verification is the fact that the state-space of a concurrent system grows exponentially both in the number of processes and in the number of data variables. To cope with the latter, the PI and a student using *abstract interpretation* to coarsen the distinctions between different data values [45]. In particular, it was shown that if an abstraction on data values is well-behaved in a precisely defined sense, then the induced process abstractions will also be well-behave in the sense that properties of the abstracted system are also properties of the concrete one.

Real-time, synchronous and probabilistic systems. The results obtained in the modeling of real-time and probabilistic systems fell into two categories: semantic issues of systems featuring such behavior, and case studies. In the case of probabilistic systems, the PI and a group of collaborators developed a semantic model based on the intuition that what characterizes such a system is the probability with which it passes tests. Using this notion we developed a semantic model of systems and a corresponding notion of refinement for such systems [32, 129].

In the case of synchronous languages the PI and two collaborators (including Gerald Luettgen) showed how the graphical design language Statecharts could be given a compositional process-algebraic semantics [93]. Based on these results, traditional techniques from process algebra such as compositional minimization can be applied to Statecharts expressions to simplify their analysis.

Regarding real-time, the semantic results focused on: the development of a model of distributed real-time systems in which different processes are governed by different clocks [36]; the definition of a semantic model for real-time in which one process refines another if it is "more predictable" [111]; and the establishment of a correspondence between timing and priority [14, 15]. The last topic also had direct practical applications, and so the following describes it in slightly more detail. The papers showed that one could treat the delays that occur before actions as being analogous to priorities for those actions: the longer an action a delays, the more likely it is to be pre-empted, and hence the lower its priority. The practical impact of this correspondence is that priority-oriented models typically are much more compact. To demonstrate the utility of this point of view the

paper presented the analysis of the SCSI-2 bus protocol; the priority-based model had an order of magnitude fewer states than the real-time model while still preserving all the timing properties of interest.

In addition to the SCSI-2 case study just discussed, the PI and two civil engineers also systematically studied the design of an active structural control system used to protect buildings against earthquakes. Using features provided by the Concurrency Workbench of North Carolina they were able to show that the system satisfied timing properties demanded of it, even though the system had in excess of 10^{19} states [58]. That the analysis was possible was due to to the semantic minimization routines provided by the tool.

Tool Development. The PI's group developed two significant pieces of software during the project. In September 1996 they released Version 1.0 of the Concurrency Workbench of North Carolina (CWB-NC), a process-algebra-based verification tool [46] that includes support for equivalence, refinement, and model checking verification methodologies. Version 1.11 is the current release (see URL `www.cs.sunysb.edu/~rance/` for details). The software has been acquired by over 350 users in 18 different countries and has been used extensively by at least three different groups in Britain and the US.

The second piece of software, the Process Algebra Compiler (PAC) [41, 47], greatly simplifies the task of retargeting the CWB-NC to new design languages. Given formal, high-level descriptions of the syntax and semantics of the language, the PAC generates the source code needed to allow the CWB-NC to process and analyze designs in the language. All six of the front ends for the current release of the tool have been generated using the PAC.

Impact on Education and Human Resources

Four PhD students and two MS students worked on various aspects of the project during its duration. In addition, three undergraduate honors students did research projects involving the CWB-NC, and a visiting PhD student from the University of Passau in Germany (Dr. Luettgen) also worked on topics related to the project. The CWB-NC has been used in graduate-level classes on verification at NCSU and at Virginia Tech and in an industrial short course taught by the PI at NCSU.

Publications

The following publications resulted from this project.

[11, 12, 13, 14, 15, 32, 35, 36, 37, 38, 39, 40, 41, 45, 46, 47, 48, 55, 56, 57, 58, 83, 93, 108, 109, 110, 111, 129]

It should be noted that this is *not* an exhaustive list of publications from the PI's group during the time frame of the proposal. A full listing of the PI's publications may be found at URL `www.cs.sunysb.edu/~rance/`.

References

[1] *ARINC Specification 659: Backplane Data Bus.* Aeronautical Radio, Inc., Annapolis, MD, December 1993.

[2] *First Workshop on Formal Methods in Software Practice*, San Diego, January 1996.

[3] R. Alur, G.J. Holzmann, and D. Peled. An analyzer for message sequence charts. *Software Concepts and Tools*, 17(2):70–77, 1996.

[4] R. Alur and M. Yannakakis. Model checking of message sequence charts. In Baeten and Mauw [6], pages 114–129.

[5] H.R. Andersen. Model checking and boolean graphs. *Theoretical Computer Science*, 126(1):3–30, April 1994.

[6] J.C.M. Baeten and S. Mauw, editors. *CONCUR '99*, volume 1664 of *Lecture Notes in Computer Science*, Eindhoven, the Netherlands, August 1999. Springer-Verlag.

[7] J.C.M. Baeten and W.P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England, 1990.

[8] H. Ben-Abdallah and S. Leue. MESA: Support for scenario-based design of concurrent systems. In B. Steffen, editor, *Fourth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS' 98)*, volume 1384 of *Lecture Notes in Computer Science*, pages 118–135, Lisbon, Portugal, March/April 1998. Springer-Verlag.

[9] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[10] G. Bhat. *Tableau-Based Approaches to Model-Checking.* PhD thesis, North Carolina State University, Raleigh, 1998.

[11] G. Bhat and R. Cleaveland. Efficient local model checking for fragments of the modal μ-calculus. In Margaria and Steffen [98], pages 107–126.

[12] G. Bhat and R. Cleaveland. Efficient model checking via the equational μ-calculus. In *Eleventh Annual Symposium on Logic in Computer Science (LICS '96)*, pages 304–312, New Brunswick, New Jersey, July 1996. IEEE Computer Society Press.

[13] G. Bhat, R. Cleaveland, and O. Grumberg. Efficient on-the-fly model checking for CTL*. In *Tenth Annual Symposium on Logic in Computer Science (LICS '95)*, pages 388–397, San Diego, July 1995. IEEE Computer Society Press.

[14] G. Bhat, R. Cleaveland, and G. Luettgen. Dynamic priorities for modeling real-time. In T. Mizuno, N. Shiratori, T. Higashino, and A. Togashi, editors, *Formal Desciption Techniques and Protocol Specification, Testing and Verification (FORTE X/PSTV XVII '97)*, pages 321–336, Osaka, November 1997. Chapman and Hall.

[15] G. Bhat, R. Cleaveland, and G. Luettgen. A practical approach to implementing real-time semantics. *Annals of Software Engineering*, 7, 1999.

[16] T. Bolognesi and E. Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks and ISDN Systems*, 14:25–59, 1987.

[17] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Object Technology Series. Addison Wesley Longman, Reading, MA, 1998.

[18] R. Breu, R. Grosu, C. Hofmann, F. Huber, I. Krueger, B. Rumpe, M. Schmidt, and W. Schwerin. Exemplary and complete object interaction descriptions. In *OOPSLA'97 Workshop on Object-oriented Behavioral Semantics*, 1997.

[19] S. D. Brookes and A. W. Roscoe. An improved failures model for communicating processes. In A. W. Roscoe S. D. Brookes and G. Winskel, editors, *Proceedings of the Seminar on Concurrency*, volume 197 of *Lecture Notes in Computer Science*, pages 281–305, Pittsburgh, Pennsylvania, July 1984. Springer-Verlag.

[20] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the Association for Computing Machinery*, 31(3):560–599, July 1984.

[21] M. Broy and I. Krueger. Interaction interfaces—toward a scientific foundation of a methodological usage of message sequence charts. In *ICFEM '98*. IEEE Computer Society Press, 1998.

[22] G. Bruns. *Distributed Systems Analysis with CCS*. Prentice-Hall, London, 1997.

[23] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170, June 1992.

[24] R.W. Butler, S.P. Miller, J.N. Potts, and V.A. Carreño. A formal methods approach to the analysis of mode confusion. In *Seventh Digital Avionics Systems Conference (DASC '98)*, Bellevue, WA, USA, November 1998. IEEE. Proccedings available on CD-ROM.

[25] U. Celikkan. *Semantic Preorders in the Automated Verification of Concurrent Systems*. PhD thesis, North Carolina State University, Raleigh, 1995.

[26] K.M. Chandy and J. Misra. *Parallel Program Design: A Foundation*. Addison-Wesley, Reading, Massachusetts, 1988.

[27] E.M. Clarke and E.A. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In D. Kozen, editor, *Logics of Programs: Workshop*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, Yorktown Heights, May 1981. Springer-Verlag.

[28] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.

[29] E.M. Clarke, O. Grumberg, and K. Hamaguchi. Another look at ltl model checking. *Formal Methods in System Design*, 10(1):47–71, February 1997.

[30] E.M. Clarke and J.M. Wing. Formal methods: state of the art and future directions. *ACM Computing Surveys*, 28(4):626–643, December 1996.

[31] R. Cleaveland. Analyzing concurrent systems using the Concurrency Workbench. In P.E. Lauer, editor, *Functional Programming, Concurrency, Simulation and Automated Reasoning*, volume 693 of *Lecture Notes in Computer Science*, pages 129–144. Springer-Verlag, 1993.

[32] R. Cleaveland, Z. Dayar, S. Smolka, and S. Yuen. Testing preorders for probabilistic processes. *Information and Computation*, to appear.

[33] R. Cleaveland and M.C.B. Hennessy. Testing equivalence as a bisimulation equivalence. In Sifakis [119], pages 11–23.

[34] R. Cleaveland and M.C.B. Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Aspects of Computing*, 5:1–20, 1993.

[35] R. Cleaveland, S. Purushothaman Iyer, and D. Yankelevich. Optimality and abstraction in model checking. In A. Mycroft, editor, *Static Analysis*, volume 983 of *Lecture Notes in Computer Science*, pages 51–63, Glasgow, UK, September 1995. Springer-Verlag.

[36] R. Cleaveland, G. Luettgen, and M. Mendler. An algebraic theory of multiple clocks. In A. Mazurkiewicz and J. Winkowski, editors, *CONCUR '97*, volume 1243 of *Lecture Notes in Computer Science*, pages 166–180, Warsaw, July 1997. Springer-Verlag.

[37] R. Cleaveland, G. Luettgen, and V. Natarajan. A process algebra with distributed priorities. In Montanari and Sassone [106], pages 34–49.

[38] R. Cleaveland, G. Luettgen, and V. Natarajan. A process algebra with distributed priorities. *Theoretical Computer Science*, 195(2):227–258, March 1998.

[39] R. Cleaveland, G. Luettgen, V. Natarajan, and S. Sims. Modeling and verifying distributed systems using priorities: A case study. *Software Concepts and Tools*, 17(2):50–62, 1996.

[40] R. Cleaveland, G. Luettgen, V. Natarajan, and S. Sims. Priorities for verifying distributed systems. In Margaria and Steffen [98], pages 278–297.

[41] R. Cleaveland, E. Madelaine, and S. Sims. A front-end generator for verification tools. In E. Brinksma, R. Cleaveland, K.G. Larsen, and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '95)*, volume 1019 of *Lecture Notes in Computer Science*, pages 153–173, Aarhus, Denmark, May 1995. Springer-Verlag.

[42] R. Cleaveland, J. Parrow, and B. Steffen. The Concurrency Workbench. In Sifakis [119], pages 24–37.

[43] R. Cleaveland, J. Parrow, and B. Steffen. A semantics-based tool for the verification of finite-state systems. In *Proceedings of the IFIP Symposium on Protocol Specification, Testing and Verification*, pages 287–302, Enschede, The Netherlands, June 1989. North-Holland.

[44] R. Cleaveland, J. Parrow, and B. Steffen. The Concurrency Workbench: A semantics-based tool for the verification of finite-state systems. *ACM Transactions on Programming Languages and Systems*, 15(1):36–72, January 1993.

[45] R. Cleaveland and J. Riely. Testing-based abstractions for concurrent systems. In Jonsson and Parrow [81], pages 417–432.

[46] R. Cleaveland and S. Sims. The NCSU Concurrency Workbench. In R. Alur and T. Henzinger, editors, *Computer Aided Verification (CAV '96)*, volume 1102 of *Lecture Notes in Computer Science*, pages 394–397, New Brunswick, New Jersey, July 1996. Springer-Verlag.

[47] R. Cleaveland and S. Sims. Generic tools for verifying concurrent systems. *Science of Computer Programming*, to appear.

[48] R. Cleaveland and S. Smolka. Strategic directions in concurrency research. *ACM Computing Surveys*, 28(4):607–625, December 1996.

[49] R. Cleaveland and B. Steffen. When is 'partial' adequate? A logic-based proof technique using partial specifications. In *Fifth Annual Symposium on Logic in Computer Science (LICS '90)*, pages 440–449, Philadelphia, June 1990. IEEE Computer Society Press.

[50] R. Cleaveland and B. Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2:121–147, 1993.

[51] C. Courcoubetis, M.Y. Vardi, P. Wolper, and M. Yannakakis. Memory efficient algorithms for verification of temporal properties. *Formal Methods in System Design*, 1:275–288, 1992.

[52] W. Damm and D. Harel. LSCs: Breathing life into message sequence charts. In A. Fantechi, P. Ciancarini, and R.Gorrieri, editors, *Third International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS' 99)*, Florence, Italy, February 1999. Kluwer Academic Publishers.

[53] R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1983.

[54] B.P. Douglass. *Real-Time UML: Developing Efficient Objects for Embedded Systems.* Object Technology Series. Addison Wesley Longman, Reading, MA, 1998.

[55] W. Elseaidy, J.W. Baugh Jr., and R. Cleaveland. Verification of an active control system using temporal process algebra. *Engineering with Computers*, 12:46–61, 1996.

[56] W. Elseaidy, R. Cleaveland, and J.W. Baugh Jr. Verifying an intelligent structure control system: A case study. In *Proceedings of the Real-Time Systems Symposium*, pages 271–275, San Juan, Puerto Rico, December 1994. IEEE Computer Society Press.

[57] W. Elseaidy, R. Cleaveland, and J.W. Baugh Jr. Formal timing analysis for fault-tolerant active structural control systems. In *First Workshop on Formal Methods in Software Practice* [2], pages 120–131.

[58] W. Elseaidy, R. Cleaveland, and J.W. Baugh Jr. Modeling and verifying active structural control systems. *Science of Computer Programming*, 29(1–2):99–122, July 1997.

[59] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. North-Holland, 1990.

[60] E.A. Emerson and J.Y. Halpern. 'Sometime' and 'not never' revisited: On branching versus linear time temporal logic. *Journal of the Association for Computing Machinery*, 33(1):151–178, January 1986.

[61] E.A. Emerson, C. Jutla, and A.P. Sistla. On model-checking for fragments of μ-calculus. In C. Courcoubetis, editor, *Computer Aided Verification (CAV '93)*, volume 697 of *Lecture Notes in Computer Science*, pages 385–396, Elounda, Greece, June/July 1993. Springer-Verlag.

[62] E.A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8:275–306, 1987.

[63] N. Francez. *Fairness*. Springer-Verlag, New York, 1986.

[64] R. Gerth, D. Peled, M. Vardi, and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Proceedings of the IFIP Symposium on Protocol Specification, Testing and Verification*, pages 3–18, Warsaw, June 1995. Chapman and Hall.

[65] P. Godefroid and P. Wolper. A partial approach to model checking. *Information and Computation*, 110(2):305–326, May 1994.

[66] R. Goering. Model checking expands verification's scope. *EE Times*, 1997. Issue 939, February 3, 1997.

[67] O. Grumberg and D.E. Long. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems*, 16(3):843–871, May 1994.

[68] B.T. Hailpern. *Verifying Concurrent Processes Using Temporal Logic*, volume 129 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1982.

[69] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.

[70] D. Harel and A. Naamad. The STATEMATE semantics of Statecharts. *ACM Transactions on Software Engineering*, 5(4):293–333, October 1996.

[71] D. Harel and M. Politi. *Modeling Reactive Systems with Statecharts: The STATEMATE Approach*. McGraw Hill, 1998.

[72] M.C.B. Hennessy. *Algebraic Theory of Processes*. MIT Press, Boston, 1988.

[73] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, London, 1985.

[74] K. Hoyme and K. Driscoll. Safebus^{tm}. *IEEE Aerospace and Electronic Systems Magazine*, 8(3):34–39, March 1993.

[75] D. Hughes and M. Dornheim. Automated cockpits: Who's in charge? *Aviation Week & Space Technology*, January 30/February 6 1995.

[76] ITU-T. Specification and description language (SDL). Recommendation Z.100, Revision 1, ITU, 1994.

[77] ITU-TS recommendation of Z.120: Message sequence chart (MSC). ITU-TS.

[78] I. Jacobson, G. Booch, and J. Rumbaugh. Excerpt from The Unified Software Development Process: The unified process. *IEEE Software*, 16(3):82–90, May/June 1999.

[79] I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software Development Process*. Object Technology Series. Addison Wesley Longman, Reading, MA, 1999.

[80] C.B. Jones. Tentative steps toward a development method for intefering programs. *ACM Transactions on Programming Languages and Systems*, 5(4):596–619, 1983.

[81] B. Jonsson and J. Parrow, editors. *CONCUR '94*, volume 836 of *Lecture Notes in Computer Science*, Uppsala, Sweden, August 1994. Springer-Verlag.

[82] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, December 1983.

[83] K. Narayan Kumar, R. Cleaveland, and S. Smolka. Infinite probabilistic and nonprobabilistic testing. In V. Arvind and R. Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 1530 of *Lecture Notes in Computer Science*, pages 209–220, Chennai, India, December 1998. Springer-Verlag.

[84] O. Kupferman and M.Y. Vardi. On the complexity of branching modular model checking. In I. Lee and S.A. Smolka, editors, *CONCUR '95*, volume 962 of *Lecture Notes in Computer Science*, pages 408–422, Philadelphia, Pennsylvania, August 1995. Springer-Verlag.

[85] P.B. Ladkin and S. Leue. Intepreting message flow graphs. *Formal Aspects of Computing*, 7(5):473–509, September/October 1995.

[86] L. Lamport. Specifying concurrent program modules. *ACM Transactions on Programming Languages and Systems*, 5:190–222, 1983.

[87] K.G. Larsen and B. Thomsen. A modal process logic. In *Third Annual Symposium on Logic in Compter Science (LICS '88)*, pages 203–210, Edinburgh, Scotland, July 1988. IEEE Computer Society Press.

[88] N.G. Leveson. *SafeWare: System Safety and Computers*. Addison Wesley Longman, Inc., Reading, Massachusetts, 1995.

[89] N.G. Leveson, M. Heimdahl, M. Hildreth, H. Reese, and J. Ortega. Experiences using Statecharts for a system requirements specification. In *Proceedings of the Sixth International Workshop on Software Specification and Design*, pages 31–41, Como, Italy, October 1991. IEEE Computer Society Press.

[90] N.G. Leveson, L.D. Pinnel, S.D. Sandys, S. Koga, and J.D. Reese. Analyzing software specifications for mode confusion potential. In *Workshop on Human Error and System Development*, Glasgow, UK, March 1997.

[91] F. Levi. *Verification of Temporal and Real-Time Properties of Statecharts*. PhD thesis, University of Pisa-Genova-Udine, Pisa, Italy, February 1997.

[92] O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Twelfth Annual ACM Symposium on Principles of Programming Languages (POPL '85)*, pages 97–107, New Orleans, Louisiana, January 1985. ACM Press.

[93] G. Luettgen, M. von der Beeck, and R. Cleaveland. Statecharts via process algebra. In Baeten and Mauw [6], pages 399–414.

[94] G. Lüttgen and V. Carreño. Murphi, SMV, and Spin models of the mode logic. See http://www.icase.edu/~luettgen/publications/publications.html#SPIN99.

[95] A. Maggiolo-Schettini, A. Peron, and S. Tini. Equivalences of Statecharts. In Montanari and Sassone [106], pages 687–702.

[96] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, Berlin, 1992.

[97] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems.* Springer-Verlag, New York, 1995.

[98] T. Margaria and B. Steffen, editors. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '96)*, volume 1055 of *Lecture Notes in Computer Science*, Passau, Germany, March 1996. Springer-Verlag.

[99] S. Mauw and M.A. Reniers. An algebraic semantics of basic message sequence charts. *The Computer Journal*, 37(4):269–277, 1994.

[100] S. Mauw and M.A. Reniers. Refinement in interworkings. In Montanari and Sassone [106], pages 671–686.

[101] E. Mikk, Y. Lakhnech, C. Petersohn, and M. Siegel. On formal semantics of Statecharts as supported by STATEMATE. In *Second BCS-FACS Northern Formal Methods Workshop*, Ilkley, UK, July 1997. Springer-Verlag.

[102] S.P. Miller. Specifying the mode logic of a flight guidance system in CoRE and SCR. In M. Ardis, editor, *Second Workshop on Formal Methods in Software Practice (FMSP '98)*, pages 44–53, Clearwater Beach, FL, USA, March 1998. ACM Press.

[103] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[104] R. Milner. *Communication and Concurrency.* Prentice-Hall, London, 1989.

[105] R. Mojdehbakhsh, W.-T. Tsai, S. Kirani, and L. Elliott. Retrofitting software safety in an implantable medical device. *IEEE Software*, pages 41–50, January 1994.

[106] U. Montanari and V. Sassone, editors. *CONCUR '96*, volume 1119 of *Lecture Notes in Computer Science*, Pisa, Italy, August 1996. Springer-Verlag.

[107] J.J. Moskwa and J.K. Hedrick. Modeling and validation of automotive engines for control algorithm development. *ASME Journal of Dynamic Systems, Measurement and Control*, 114(2):278–285, June 1992.

[108] M. Mueller-Olm, B. Steffen, and R. Cleaveland. On the evolution of reactive components—a process-algebraic approach. In J.-P. Finance, editor, *Fundamental Approaches to Software Engineering*, volume 1577 of *Lecture Notes in Computer Science*, pages 161–175, Berlin, March 1999. Springer-Verlag.

[109] M. Narasimha, R. Cleaveland, and P. Iyer. Probabilistic temporal logics via the modal mu-calculus. In W. Thomas, editor, *Foundations of Software Science and Computation Structures*, volume 1578 of *Lecture Notes in Computer Science*, pages 288–305, Amsterdam, March 1999. Springer-Verlag.

[110] V. Natarajan and R. Cleaveland. Divergence and fair testing. In Z. Fülöp and F. Gécseg, editors, *Automata, Languages and Programming (ICALP '95)*, volume 944 of *Lecture Notes in Computer Science*, pages 648–659, Szeged, Hungary, July 1995. Springer-Verlag.

[111] V. Natarajan and R. Cleaveland. Predictability of real-time systems: A process-algebraic approach. In *Proceedings of the Real-Time Systems Symposium*, pages 82–91, Washington, DC, December 1996. IEEE Computer Society Press.

[112] K. Norrie and P. Curran. Using formal methods to enhance the quality of a standard for medical device communications. In *First Workshop on Formal Methods in Software Practice* [2], pages 132–140.

[113] D. Peled. A toolset for message sequence charts. In A.J. Hu and M.Y. Vardi, editors, *Tenth International Conference on Computer Aided Verification (CAV' 98)*, number 1427 in Lecture Notes in Computer Science, pages 532–536, Vancouver, BC, Canada, June 1998.

[114] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13(1):45–60, January 1981.

[115] J.P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In M. Dezani-Ciancaglini and U. Montanari, editors, *Proceedings of the International Symposium in Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351, Turin, April 1982. Springer-Verlag.

[116] T. Rathje and S. Sandler. CPU formal verification receives a boost. *EE Times*, 1996. Issue 927, November 11, 1996.

[117] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Refrence Manual.* Object Technology Series. Addison Wesley Longman, Reading, MA, 1999.

[118] F.B. Schneider. *On Concurrent Programming.* Springer-Verlag, 1997.

[119] J. Sifakis, editor. *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lecture Notes in Computer Science*, Grenoble, June 1989. Springer-Verlag.

[120] S. Sims. *Customizable Tools for Verifying Concurrent Systems.* PhD thesis, North Carolina State University, Raleigh, 1997.

[121] C. Stirling. Modal and temporal logics. In S. Abramsky, D. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 477–563. Oxford University Press, 1992.

[122] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 133–191. North-Holland, 1990.

[123] A.C. Uselton and S.A. Smolka. A compositional semantics for Statecharts using labeled transition systems. In Jonsson and Parrow [81], pages 2–17.

[124] A.C. Uselton and S.A. Smolka. A process-algebraic semantics for Statecharts via state refinement. In *IFIP TC2 Working Conference on Programming Concepts, Methods and Calculi (PROCOMET '94)*. North Holland/Elsevier, 1994.

[125] M. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Symposium on Logic in Computer Science (LICS '86)*, pages 332–344, Cambridge, Massachusetts, June 1986. IEEE Computer Society Press.

[126] D.J. Walker. Bisimulation and divergence in CCS. In *Third Annual Symposium on Logic in Computer Science*, pages 186–192, Edinburgh, Scotland, July 1988. IEEE Computer Society Press.

[127] J.B. Warmer and A.G. Kleppe. *The Object Constraint Language: Precise Modeling With UML*. Object Technology Series. Addison Wesley Longman, Reading, MA, 1999.

[128] J.B. Warmer and A.G. Kleppe. OCL: The constraint language of the UML. *Journal of Object-Oriented Programming*, 12(1):10–13,28, March 1999.

[129] S. Yuen, R. Cleaveland, Z. Dayar, and S. Smolka. Fully abstract characterizations of testing preorders for probabilistic processes. In Jonsson and Parrow [81], pages 497–512.

WALTER RANCE CLEAVELAND II
Professor of Computer Science

Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400

Tel: (516) 632-8448 (voice), (516) 632-8334 (fax)
E-mail: `rance@cs.sunysb.edu`
URL: `www.cs.sunysb.edu/~rance`

Research interests: Automated and interactive tools for reasoning about computer systems. Specification and verification of concurrent and distributed systems. Semantics of programming languages and logics. Applications of logic in Computer Science.

Postgraduation work experience:

From 1998 Professor, Computer Science Department, SUNY at Stony Brook.

1994–1998 Associate Professor, Computer Science Department, N.C. State University.

1989–1994 Assistant Professor, Computer Science Department, N.C. State University.

1987–1989 Research Associate, Computer Science Department, Sussex University.

Education:

Cornell University: PhD May 1987, MS June 1985 in Computer Science

Duke University: BS *summa cum laude* May 1982 in Mathematics and Computer Science

Awards:

1997 Elected Member of IFIP Working Group 2.2

1994 Alcoa Foundation Research Achievement Award

1992 NSF National Young Investigator

1992 ONR Young Investigator

1991 Shell Undergraduate Teaching Award

Five publications most relevant to project:

1. G. Luettgen, M. von der Beeck, and R. Cleaveland. Statecharts via process algebra. In J.C.M. Baeten and S. Mauw, editors, *Tenth International Conference on Concurrency Theory (CONCUR '99)*, pages 399-414, volume 1664 of *Lecture Notes in Computer Science*, Eindhoven, The Netherlands, August 1999. Springer-Verlag.

2. G. Bhat and R. Cleaveland. Efficient model checking via the equational mu-calculus. In 11^{th} *Annual Symposium on Logic in Computer Science*, pages 304–312, New Brunswick, New Jersey, July 1996. IEEE Computer Society Press.

3. R. Cleaveland, J. Parrow, and B. Steffen. The Concurrency Workbench: A semantics-based verification tool for finite-state concurrent systems. *ACM Transactions on Programming Languages and Systems*, 15(1):36–72, 1993.

4. R. Cleaveland, G. Luettgen, V. Natarajan, and S. Sims. Modeling and verifying distributed systems using priorities: A case study. *Software Concepts and Tools*, 17(2):50–62, 1996.

5. M. Mueller-Olm, B. Steffen, and R. Cleaveland On the Evolution of Reactive Components: A Process-Algebraic Approach. In J.-P. Finance, editor, *Fundamental Approaches to Software Engineering*, volume 1577 of *Lecture Notes in Computer Science*, pages 161-175, Amsterdam, March 1999. Springer-Verlag.

Five other publications:

1. G. Bhat, R. Cleaveland, and O. Grumberg. Efficient on-the-fly model checking for CTL*. In *Tenth Annual Symposium on Logic in Computer Science (LICS '95)*, pages 388–397, San Diego, July 1995. IEEE Computer Society Press.

2. G. Bhat, R. Cleaveland, and G. Luettgen. A practical approach to implementing real-time semantics. *Annals of Software Engineering*, 7, 1999. To appear.

3. R. Cleaveland, G. Luettgen, and V. Natarajan. A process algebra with distributed priorities. *Theoretical Computer Science*, 195(2):227–258, 1998.

4. R. Cleaveland and S. Smolka Strategic directions in concurrency research. *ACM Computing Surveys* 28(4):607–625, December 1996. (Special issue on *ACM Strategic Directions in Computing Research* workshop.)

5. W. Elseaidy, R. Cleaveland, and J. Baugh. Modeling and verifying active structural control systems. *Science of Computer Programming*, 29(1–2):99–122, July 1997.

Past and *present* graduate students: Girish Bhat (PhD NCSU, 1998), Neerja Bhatt (MS NCSU, 1996), Ufuk Celikkan (PhD NCSU, 1995), Zeynep Dayar (MS NCSU, 1997), Andre Fredette (PhD NCSU, 1993), Jayesh Gada (MS NCSU, 1995), Sunil Jain (MS NCSU, 1994), *Tan Li* (PhD Stony Brook, expected 2002), Gerald Luettgen (PhD University of Passau, 1998) Granville Miller (MS NCSU, 1993), Bradford Mott (MS NCSU, 1997), V. Natarajan (PhD NCSU, 1996), James Riely (PhD UNC, 1999), *Bikram Sengupta* (PhD Stony Brook, expected 2002), Steven Sims (PhD NCSU, 1997), Pranav Tiwari (MS NCSU, 1997), Vikas Trehan (MS NCSU, 1992), Yutao Xie (MS NCSU, 1998).

Graduate advisor: Robert L. Constable (Cornell University)

Other collaborators: Michael von der Beeck (Technical University of Munich, Germany) Marco Bernardo (University of Bologna, Italy), Ivan Christoff (Uppsala University, Sweden), Orna Grumberg (The Technion, Israel), Matthew Hennessy (University of Sussex, England), Insup Lee (University of Pennsylvania), Phil Lewis (SUNY at Stony Brook), Eric Madelaine (INRIA-Sophia Antipolis, France), Markus Mueller-Olm (University of Dortmund, Germany), Murali Narasimha (Ericsson, Raleigh, North Carolina), S. Purushothaman Iyer (North Carolina State University), Scott Smolka (SUNY at Stony Brook), Oleg Sokolsky (University of Pennsylvania), Bernhard Steffen (University of Dortmund, Germany), Shoji Yuen (Nagoya University, Japan).

GERALD LUETTGEN
Staff Scientist of Computer Science

Institute for Computer Applications in Science and Engineering (ICASE)
NASA Langley Research Center
Hampton, VA 23681-2199

Phone: (757) 864-8003
Fax: (757) 864-6134
E-mail: `luettgen@icase.edu`
URL: `www.icase.edu/~luettgen`

Research interests: Formal techniques for the specification, analysis, and verification of concurrent and distributed systems. Semantics of specification/programming languages and logics.

Postgraduate work experience:

- *July 1998 – present:* Staff Scientist. Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, Virginia.
- *August 1994 – June 1998:* Staff Scientist. Department of Mathematics and Computer Science, University of Passau, Germany.

Education:

- *May 1998:* Doctoral Degree in Natural Sciences, Department of Mathematics and Computer Science, University of Passau, Germany.
- *July 1994:* Diploma in Computer Science, Department of Computer Science, Aachen University of Technology, Germany.

Awards:

- Nominated for the 1998 Dissertation Award of the German Society of Computer Scientists (Gesellschaft für Informatik, GI); only nominee from the University of Passau, Germany.
- Winner of a doctoral grant from the German Academic Exchange Service (DAAD) for a visit (April 1995 – March 1996) to North Carolina State University, Raleigh, North Carolina.

Five publications most relevant to the project:

1. G. Luettgen, M. von der Beeck, and R. Cleaveland. Statecharts via process algebra. In J.C.M. Baeten and S. Mauw, editors, *Tenth International Conference on Concurrency Theory (CONCUR '99)*, pages 399-414, volume 1664 of *Lecture Notes in Computer Science*, Eindhoven, The Netherlands, August 1999. Springer-Verlag.
2. G. Luettgen and V. Carreño. Analyzing mode confusion via model checking. In D. Dams, R. Gerth, S. Leue, and M. Massink, editors, *Theoretical and Practical Aspects of SPIN Model Checking* (SPIN '99), volume 1680 of *Lecture Notes in Computer Science*, pages 120–135, Toulouse, France, September 1999. Springer-Verlag.
3. R. Cleaveland, G. Luettgen, and V. Natarajan. A process algebra with distributed priorities. *Theoretical Computer Science*, 195(2):227–258, 1998.

4. G. Bhat, R. Cleaveland, and G. Luettgen. A practical approach to implementing real-time semantics. *Annals of Software Engineering*, 7, 1999. To appear.

5. R. Cleaveland, V. Natarajan, S. Sims, and G. Luettgen. Modeling and verifying distributed systems using priorities: A case study. *Software–Concepts and Tools*, 17(2):50–62, 1996.

Five other publications:

1. R. Cleaveland, G. Luettgen, and M. Mendler. An algebraic theory of multiple clocks. In A. Mazurkiewicz and J. Winkowski, editors, *Eighth International Conference on Concurrency Theory (CONCUR '97)*, volume 1243 of *Lecture Notes in Computer Science*, pages 166–180, Warsaw, Poland, July 1997. Springer-Verlag.

2. R. Cleaveland, G. Luettgen, and V. Natarajan. Priority in process algebra. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*. Elsevier Science Publishers, 1999. To appear.

3. S. Graf, B. Steffen, and G. Luettgen. Compositional minimisation of finite state systems using interface specifications. *Formal Aspects of Computing*, 8(5):607–616, 1996.

4. A. Geser, J. Knoop, G. Luettgen O. Rüthing, and B. Steffen. Non-monotone fixpoint iterations to resolve second order effects. In T. Gyimóthy, editor, *Sixth International Symposium on Compiler Construction (CC '96)*, volume 1060 of *Lecture Notes in Computer Science*, pages 106–120, Linköping, Sweden, April 1996. Springer-Verlag.

5. R. Cleaveland, G. Luettgen, and M. Mendler. An algebraic theory of multiple clocks. In A. Mazurkiewicz and J. Winkowski, editors, *CONCUR '97*, volume 1243 of *Lecture Notes in Computer Science*, pages 166–180, Warsaw, July 1997. Springer-Verlag.

Graduate student: Marco Kick (University of Passau, Germany).

Graduate advisors: Rance Cleaveland (SUNY at Stony Brook, New York), Bernhard Steffen (University of Dortmund, Germany).

Other collaborators: Girish Bhat (Make Systems Inc., Cary, North Carolina), Ricky Butler (NASA Langley Research Center, Hampton, Virginia), Victor Carreño (NASA Langley Research Center, Hampton, Virginia), Gianfranco Ciardo (College of William and Mary, Williamsburg, Virginia), Rance Cleaveland (SUNY at Stony Brook, New York), Ben Di Vito (NASA Langley Research Center, Hampton, Virginia), Michael Mendler (University of Sheffield, England), Paul Miner (NASA Langley Research Center, Hampton, Virginia), César Muñoz (Insitute for Computer Applications in Science and Engineering, Hampton, Virginia), Vaidhyanathan Natarajan (IBM Corporation, Research Triangle Park, North Carolina), Radu Siminiceanu (College of William and Mary, Williamsburg, Virginia), Bernhard Steffen (University of Dortmund, Germany), Michael von der Beeck (Technology University of Munich, Germany).

SUMMARY PROPOSAL BUDGET YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
SUNY at Stony Brook

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
W. Rance Cleaveland

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **W. Rance Cleaveland - Professor**	**0.00**	**0.00**	**2.00**	$ **19,550**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**0.00**	**0.00**	**2.00**	**19,550**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**2**) GRADUATE STUDENTS				**36,000**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**55,550**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**5,386**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**60,936**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**5,000**	
2. FOREIGN				**2,500**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**5,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**19,968**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**24,968**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**93,404**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **47.5% of MTDC (Rate: 47.5000, Base: 93404)**					
TOTAL INDIRECT COSTS (F&A)				**44,366**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**137,770**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **137,770**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
W. Rance Cleaveland				
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 1

**** E- Travel**
Funds are requested to attend two domestic (assumed cost: $1500/trip) and one international (assumed cost: $2500/trip)
conference each year of the contract. In addition, as the project involves a collaboration with another institution (ICASE), funds are requested (assumed cost: $1000/trip/individual) for the PI and/or graduate students to visit ICASE.

SUMMARY PROPOSAL BUDGET

YEAR 2

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
SUNY at Stony Brook

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
W. Rance Cleaveland

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **W. Rance Cleaveland - Professor**	**0.00**	**0.00**	**2.00**	$ **20,332**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**0.00**	**0.00**	**2.00**	**20,332**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**2**) GRADUATE STUDENTS				**37,440**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**57,772**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**5,890**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**63,662**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**5,000**	
2. FOREIGN				**2,500**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**1,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**20,471**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**21,471**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**92,633**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **47.5% of MTDC (Rate: 47.5000, Base: 77194)**					
TOTAL INDIRECT COSTS (F&A)				**36,667**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**129,300**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **129,300**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
W. Rance Cleaveland		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 2

**** E- Travel**
Funds are requested to attend two domestic (assumed cost:
$1500/trip) and one international (assumed cost: $2500/trip)
conference each year of the contract. In addition, as the project
involves a collaboration with another institution (ICASE), funds are
requested (assumed cost: $1000/trip/individual) for the PI
and/or graduate students to visit ICASE each each year

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
SUNY at Stony Brook

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
W. Rance Cleaveland

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **W. Rance Cleaveland - Professor**	**0.00**	**0.00**	**2.00**	$ **21,145**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**0.00**	**0.00**	**2.00**	**21,145**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**2**) GRADUATE STUDENTS				**38,938**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**60,083**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**6,426**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**66,509**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**5,000**	
2. FOREIGN				**2,500**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**1,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**20,996**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**21,996**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**96,005**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **47.5% of MTDC (Rate: 47.5000, Base: 75009)**					
TOTAL INDIRECT COSTS (F&A)				**35,629**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**131,634**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **131,634**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
W. Rance Cleaveland		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 3

**** E- Travel**
Funds are requested to attend two domestic (assumed cost: $1500/trip) and one international (assumed cost: $2500/trip) conference each year of the contract. In addition, as the project involves a collaboration with another institution (ICASE), funds are requested (assumed cost: $1000/trip/individual) for the PI and/or graduate students to visit ICASE each each year.

SUMMARY PROPOSAL BUDGET

Cumulative

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
SUNY at Stony Brook

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
W. Rance Cleaveland

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **W. Rance Cleaveland - Professor**	**0.00**	**0.00**	**6.00**	$ **61,027**	$
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**0.00**	**0.00**	**6.00**	**61,027**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**6**) GRADUATE STUDENTS				**112,378**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**173,405**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**17,702**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**191,107**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**15,000**	
2. FOREIGN				**7,500**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**7,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**61,435**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**68,435**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**282,042**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				**116,663**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**398,705**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **398,705**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
W. Rance Cleaveland		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

Budget Justification

Faculty salary: Two months of summer support are requested for the PI for each year of the contract. A 4% annual increase is assumed.

Graduate students: Support is requested for two graduate students for each year of the project. The requested sum includes 1/2-time support during the academic year and full-time support during the summer. A 4% annual increase is assumed.

Fringe benefits: Rates are 16.5graduate students for the first year.

Travel: Funds are requested to attend two domestic (assumed cost: $1500/trip) and one international (assumed cost: $2500/trip) conference each year of the contract relevant conferences include FOSSACS, Process Algebra and Performance Modeling, CONCUR, LICS, and Computer-Aided Verification. The research area of this proposal is well-represented in Europe, and keeping abreast of latest developments in Europe is essential. In addition, as the project involves a collaboration with another institution (ICASE), funds are requested (assumed cost: $1000/trip/individual) for two of the three personnel (PI and graduate students) to visit ICASE each each year.

Materials and Supplies: The requested funds are to pay for incidental expenses associated with the research, including presentation materials. In addition, funds are requested in the first year for a workstation for the graduate student most involved with tool development to use. The graduate student computing facilities at Stony Brook are centered around different laboratories; students generally do not have machines on their desks. The project includes significant system development and usage, however, and the student in question will need a machine of his or her own in order for the work to be conducted and completed in a timely manner.

Subcontract: The subcontract is with ICASE and includes a request for two months/year of support for Dr. Gerald Luettgen, together with $1500/year of domestic travel support for Dr. Luettgen to visit Stony Brook. ICASE has agreed to provide an additional two months/year of release time for Dr. Luettgen to work on the project and is also matching the travel request.

Indirect costs: The Stony Brook indirect cost rate is 47.5The first $25,000 of the subcontract, including the entire first year subcontract amount and $5,032 of the second year, will also be billed at this rate.

SUMMARY PROPOSAL BUDGET YEAR 1

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Universities Space Research Association

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Gerald M Luettgen

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Gerald M Luettgen**	**2.00**	**0.00**	**0.00**	$ **8,656**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**2.00**	**0.00**	**0.00**	**8,656**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**0**) GRADUATE STUDENTS				**0**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**8,656**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**3,803**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**12,459**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**1,500**	
2. FOREIGN				**0**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**13,959**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **See note. (Rate: 100.0000, Base: 6009)**					
TOTAL INDIRECT COSTS (F&A)				**6,009**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**19,968**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **19,968**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
Gerald M Luettgen		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 1

**** E- Travel**
Funds are requested for the senior person to visit SUNY at Stony Brook as part of collaboration.
**** I- Indirect Costs**
USRA's overhead calculation is confidential.
A separate budget is being sent to NSF by USRA directly with a full accounting of it's overhead calculation.

SUMMARY PROPOSAL BUDGET YEAR 2

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Universities Space Research Association

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Gerald M Luettgen

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Gerald M Luettgen**	**2.00**	**0.00**	**0.00**	$ **8,916**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**2.00**	**0.00**	**0.00**	**8,916**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**0**) GRADUATE STUDENTS				**0**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**8,916**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**3,912**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**12,828**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**1,500**	
2. FOREIGN				**0**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**14,328**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **See note. (Rate: 100.0000, Base: 6143)**					
TOTAL INDIRECT COSTS (F&A)				**6,143**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**20,471**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **20,471**	$

M. COST SHARING PROPOSED LEVEL $ **0** AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
Gerald M Luettgen		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 2

**** E- Travel**
Funds are requested for the senior person to visit SUNY at Stony Brook as part of collaboration.
**** I- Indirect Costs**
USRA's overhead calculation is confidential. A separate budget is being sent to NSF by USRA directly with a full accounting of it's overhead calculation.

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Universities Space Research Association

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Gerald M Luettgen

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Gerald M Luettgen**	**2.00**	**0.00**	**0.00**	$ **9,183**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**2.00**	**0.00**	**0.00**	**9,183**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**0**) GRADUATE STUDENTS				**0**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**9,183**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**4,025**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**13,208**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**1,500**	
2. FOREIGN				**0**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**14,708**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **See note. (Rate: 100.0000, Base: 6288)**					
TOTAL INDIRECT COSTS (F&A)				**6,288**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**20,996**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **20,996**	$

M. COST SHARING PROPOSED LEVEL $ **0** | AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
Gerald M Luettgen				
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked	Date Of Rate Sheet	Initials - ORG

SUMMARY PROPOSAL BUDGET COMMENTS - Year 3

**** E- Travel**
Funds are requested for the senior person to visit SUNY at Stony Brook as part of collaboration.
**** I- Indirect Costs**
USRA's overhead calculation is confidential. A separate budget is being sent to NSF by USRA directly with a full accounting of it's overhead calculation.

SUMMARY PROPOSAL BUDGET

Cumulative

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months) Proposed	Granted
AWARD NO.		

ORGANIZATION
Universities Space Research Association

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Gerald M Luettgen

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-mos. CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. **Gerald M Luettgen**	**6.00**	**0.00**	**0.00**	$ **26,755**	$
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	**0.00**	**0.00**	**0.00**	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	**6.00**	**0.00**	**0.00**	**26,755**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL ASSOCIATES	**0.00**	**0.00**	**0.00**	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	**0.00**	**0.00**	**0.00**	**0**	
3. (**0**) GRADUATE STUDENTS				**0**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**26,755**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**11,740**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**38,495**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				**0**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				**4,500**	
2. FOREIGN				**0**	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ **0**					
2. TRAVEL **0**					
3. SUBSISTENCE **0**					
4. OTHER **0**					
(**0**) TOTAL PARTICIPANT COSTS				**0**	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				**0**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**0**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**0**	
5. SUBAWARDS				**0**	
6. OTHER				**0**	
TOTAL OTHER DIRECT COSTS				**0**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**42,995**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				**18,440**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**61,435**	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.D.7.j.)				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **61,435**	$

M. COST SHARING PROPOSED LEVEL $ **0** | AGREED LEVEL IF DIFFERENT $

PI / PD TYPED NAME & SIGNATURE*	DATE	FOR NSF USE ONLY
Gerald M Luettgen		INDIRECT COST RATE VERIFICATION
ORG. REP. TYPED NAME & SIGNATURE*	DATE	Date Checked / Date Of Rate Sheet / Initials - ORG

Current and Pending Support

(See GPG Section II.D.8 for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Other agencies (including NSF) to which this proposal has been/will be submitted.

Investigator: **W. Rance Cleaveland**

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Specification Formalisms for Component-Based Concurrent Systems**

Source of Support: **NSF**
Total Award Amount: $ **148,000** Total Award Period Covered: **06/01/98 - 05/31/00**
Location of Project: **SUNY at Stony Brook**
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr: **2.00**

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Abstraction-Based Approaches to Correct Reactive Software**

Source of Support: **ARO**
Total Award Amount: $ **270,000** Total Award Period Covered: **06/01/98 - 05/31/01**
Location of Project: **North Carolina State University**
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr: **1.00**

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Development and Implementation of Heterogeneous Verification Methods for Distributed Systems**

Source of Support: **NSF**
Total Award Amount: $ **16,395** Total Award Period Covered: **04/01/97 - 03/31/99**
Location of Project: **SUNY at Stony Brook**
Person-Months Per Year Committed to the Project. Cal: Acad: **0.50** Sumr:

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Heterogeneous Specification Formalisms for Reactive Systems**

Source of Support: **NSF**
Total Award Amount: $ **398,705** Total Award Period Covered: **05/01/00 - 04/30/03**
Location of Project: **SUNY at Stony Brook**
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr: **2.00**

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: **Automated Analysis of Probabilistic Systems**

Source of Support: **NSF**
Total Award Amount: $ **226,793** Total Award Period Covered: **05/01/00 - 04/30/03**
Location of Project: **SUNY at Stony Brook**
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr: **2.00**

---

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Current and Pending Support

(See GPG Section II.D.8 for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: **Gerald Luettgen**

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: **Heterogeneous Specification Formalisms for Reactive Systems**

Source of Support: **NSF via SUNY at Stony Brook**
Total Award Amount: $ **398,705** Total Award Period Covered: **05/01/00 - 04/30/03**
Location of Project: **ICASE**
Person-Months Per Year Committed to the Project. Cal: **4.00** Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title:

Source of Support:
Total Award Amount: $ Total Award Period Covered:
Location of Project:
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title:

Source of Support:
Total Award Amount: $ Total Award Period Covered:
Location of Project:
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title:

Source of Support:
Total Award Amount: $ Total Award Period Covered:
Location of Project:
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title:

Source of Support:
Total Award Amount: $ Total Award Period Covered:
Location of Project:
Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

H Facilities, Equipment and Other Resources

H.1 SUNY at Stony Brook

The computing facilities currently available at Stony Brook for the use of this project are housed in the Computer Science Building and include 85 Sun Sparcstations (with 5 Sparc10's, 4 Sparc LX's, 26 IPC/Sparc1+'s, and a 4-processor 512MB Sparc1000), four HP 400S TurboVRX workstations, four SGI workstations (a dual-processor Onyx, an Indigo Extreme, two 4D/25 Personal Irises, and a 4-processor 4D/240GTX) and a 10-CPU Sequent S27.

Moreover, the department has recently acquired a dual-processor, 300 MHz, UltraSparc II with 2 GB of ram. This machine was purchased with funds from an NSF Experimental Software Systems (ESS) grant—the topic of which is Model Checking and Logic Programming—and with matching funds from the University. The UltraSparc is primarily intended for verification research and will hence see significant use in the proposed effort. The amount of RAM on the machine will be upgraded to 8 GB over the next year.

The department has also acquired, under NSF infrastructure grant CDA-9303181, 15 4-processor Sparc20's and a 16-processor SGI Challenge with 3GB of RAM, 16GB of SCSI disk. These machines are housed in the PROUD (Parallel Resources on Users' Desks) laboratory and are connected by a dedicated 100 Mb/s fast Ethernet. An ATM switch connects the SGI Challenge to several other PROUD workstations.

The machines within the department are distributed over 12 subnetworks, all 10Mb/s Ethernet, tied to an Ethernet switch with a total bandwidth of 2.6Gb/s. There are also two ATM connections to a campus-wide ATM switch. The campus has a T3-link (approx. 45Mb/s) to the Internet.

H.2 ICASE

The ICASE computing facilities available to conduct the proposed research consist of a large array of state-of-the-art SUN workstations, including three Ultra 5, six Ultra 2 and approximately 50 Sparc 10s and Sparc 20s, as well as various SGI workstations, all of which are internally connected via (Fast) Ethernet and FDDI. ICASE also operates a 64-node Beowulf-style PC cluster with 400 MHz Pentium II / 500 MHz Pentium III processors and 20 GB of memory. Parallel supercomputing facilities are also available to ICASE researchers through various NASA programs, as well as arrangements with other supercomputer centers, DOE laboratories, and vendors. ICASE computing facilities are connected to NASA's fast inter-center networks and to commercial Internet with direct, full-time access via two T1 connections.