DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
SEATTLE REGIONAL OFFICE
1107 NE 45TH STREET, SUITE 350
SEATTLE WA 98105-4631

IN REPLY REFER TO:

4330
ONR 247
11 Jul 97

From: Director, Office of Naval Research, Seattle Regional Office, 1107 NE 45th St., Suite 350, Seattle, WA 98105

To: Defense Technical Center, Attn: P. Mawby, 8725 John J. Kingman Rd., Suite 0944, Ft. Belvoir, VA 22060-6218

Subj: RETURNED GRANTEE/CONTRACTOR TECHNICAL REPORTS

1. This confirms our conversations of 27 Feb 97 and 11 Jul 97. Enclosed are a number of technical reports which were returned to our agency for lack of clear distribution availability statement. This confirms that all reports are unclassified and are "APPROVED FOR PUBLIC RELEASE" with no restrictions.

2. Please contact me if you require additional information. My e-mail is *silverr@onr.navy.mil* and my phone is (206) 625-3196.

ROBERT J. SILVERMAN

To: Regional Director
Team Leader
ACO

This technical report was sent to me by DTIC because it does not include the DD-1498 form with the proper disclosure/distribution statement.

Please obtain this form with proper instructions and return it and the technical report directly to DTIC.

Also implement procedures with the contractor to correct this problem.

Thank You,

Jim Carbonara

Jim Carbonara,
Director, Field Operations

Kestrel Institute

17 November 1995

Dr Ralph Wachter Code 1133
Office of Naval Research
Ballston Tower One
800 N Quincy St
Arlington VA 22217-5660

Dear Ralph,

Find enclosed the final report for Contract N00014-93-C-0056 covering the period 1 Oct 92 - 30 Sep 95.

Sincerely,

Douglas R. Smith

enclosure

3260 Hillview Avenue, Palo Alto, California 94304 (415) 493-6871

Kestrel Institute

Foundations of Software Development

Final Report

by

Douglas R. Smith

November 1995

Prepared for:

Office of Naval Research
Computer Science Division, Code 1133
800 North Quincy Street
Arlington, Virginia 22217

Kestrel Institute • 3260 Hillview Avenue • Palo Alto, CA 94304 • (415) 493-6871

P.I. Name: Douglas R. Smith
Insitution: Kestrel Institute
Telephone: (415) 493-6871
E-mail: smith@kestrel.edu
Contract Title: Foundations of Software Development
Contract Number: N00014-93-C-0056
Reporting Period: 1 Oct 92 - 30 Sep 95

1. Productivity Measures

Refereed papers submitted but not published: 3

Refereed papers published: 12

Unrefereed reports and articles: 3

Books or parts thereof submitted but not yet published: 0

Books or parts thereof published: 4

Patents filed but not yet granted: 0

Patents granted: 0

Invited presentations: 28

Contributed presentations: 12

Honors received: 10

1. Editorial board member, "Journal of Artificial Intelligence Research".
2. Editorial Board member, "Automated Software Engineering, The International Journal of Automated Reasoning and Artificial Intelligence in Software Engineering".
3. Appointed Lecturer in Computer Science, Spring Quarters 1993-95, Computer Science Department, Stanford University.
4. Elected Chairman, IFIP Working Group 2.1 (Algorithmic Languages and Calculi), 14 May 1993.
5. Invited Speaker, LOPSTR '93 (Logic Program Synthesis Conference 1993), Louvain-la-Neuve, Belgium, 7-9 July 1993
6. Program Committee member and Session chair, Eighth Knowledge-Based Software Engineering Conference, Chicago, IL, Sept. 1993.
7. Co-organizer, Workshop on Logical Theory for Program Construction, Schloss Dagstuhl, Germany, 7-11 March 1994.
8. Served as General Chair and Program Committee member for Ninth Knowledge-Based Software Engineering Conference, Monterey, CA, 20-23 September 1994.

9. Elected Fellow of the American Association of Artificial Intelligence, April 1995 (awarded at IJCAI in Montreal, August 1995).

10. Joined Editorial Board, CONSTRAINTS, An International Journal, published by Kluwer Academic Publishers.

Prizes or awards received: 1

1. Best Paper Award, Eighth Knowledge-Based Software Engineering Conference, Chicago, IL, Sept. 1993.

Promotions obtained: 0

Graduate students supported: 0

Post-docs supported: 0

Minorities supported: 0

P.I. Name: Douglas R. Smith
Insitution: Kestrel Institute
Telephone: (415) 493-6871
E-mail: smith@kestrel.edu
Contract Title: Foundations of Software Development
Contract Number: N00014-93-C-0056
Reporting Period: 1 Oct 92 - 30 Sep 95

2. Summary of Technical Results

Algorithms and data structures are among the primary constituents of computer software and thus are among basic objects of study in Computer Science. This project is concerned with the structure and automated design of algorithms and data structures. Our scientific hypothesis is that there exist general algorithm, data structure, and design concepts that underlie and explain most of the detailed structure of conventional software systems. By abstracting and formalizing these concepts and showing how to mechanize their application, we can prepare the way for the coming generation of automated software design environments.

Our approach involves identifying classes of algorithms that solve a broad range of useful problems. In particular we have emphasized formalizing abstract algorithms that make minimal assumptions about the structure of a problem. Once a class of algorithms has been identified we represent its essence as a theory, called an *algorithm theory* [9]. Under ONR support we have developed algorithm theories and design tactics for divide-and-conquer [2], simple problem reduction [2], global search (binary search, backtrack, branch-and-bound) [3], problem reduction generators (dynamic programming, generalized branch-and-bound, game tree search) [4], local search [1], constraint propagation [12, 7, 11], and others. These have all been at least partially implemented and tested in the KIDS system [8]. KIDS has been used to derive over 70 algorithms.

More recent work has focused on theories and operations on theories as the formal underpinings of algorithm design as well as data structure design and refinement and general software development. Algorithm design is based on constructing a theory morphism[1] from an algorithm theory into a given application domain theory. Datatype design and and refinement are also based on constructing a theory morphism from an one datatype theory into another. Generally, specifications are theories and the implementation of specifications is based on constructing a theory morphism into a (relatively) concrete, computationally-oriented theory. This formal view of software development has motivated research into the kinds of theories that are useful for specifying and reasoning about application domains and systems, as well as capturing knowledge about algorithms, data structures, and other kinds of programming knowledge. It has also led us to focus our attention on formal/automatable techniques for constructing theory morphisms.

Project results during the past three years are listed below.

1. *Classification Approach to Design* –

   We developed the theoretical foundations needed to support a classification approach to software design: a declarative statement of a problem (e.g a transportation scheduling problem)

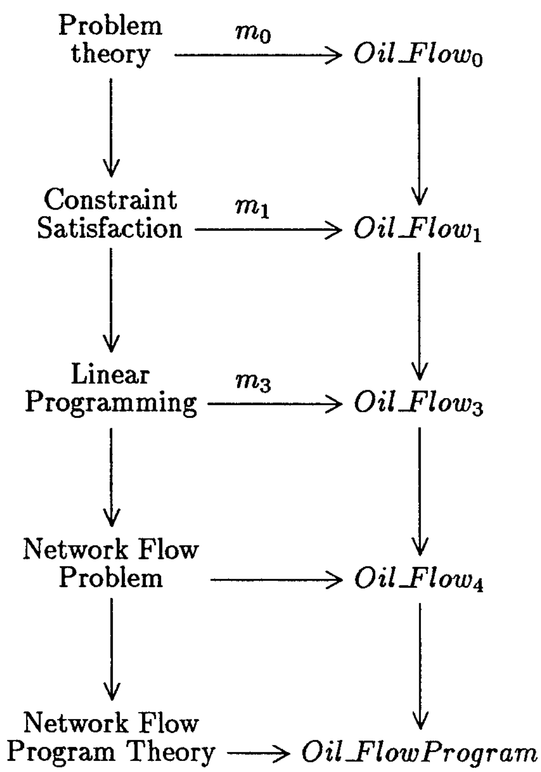
[1] A *theory morphism* from theory A to theory B is a translation of the language of A to the language of B such that theorems of A translate to theorems of B.

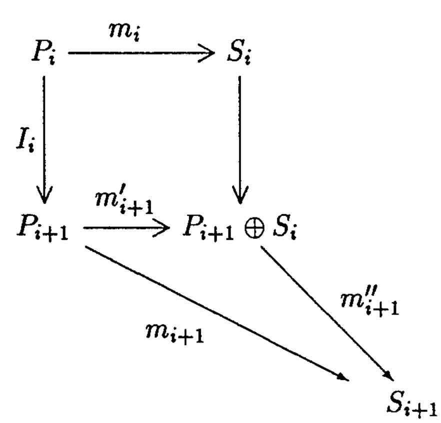
is classified with respect to a hierachic library of problem classes. Each problem class has one or more problem-solving methods associated with it. Classification exposes the implicit structure of the problem that can be exploited by a problem-solver. Thus a problem-solving method that applies to a given problem is obtained as a by-product of the classification process.

Problem-solving knowledge is represented as formal theories and arranged in a refinement hierarchy. A given problem is classified by developing morphisms from the library problem-solving theory and the given problem domain theory. The views can be constructed incrementally by starting at the root of the hierarchy and developing views one level at a time. We have discovered four basic techniques for constructing morphisms [6].

We have been implementing a new theory-based system to support this approach to design, called Specware, which has come to be the main research system under development at Kestrel. KIDS is being phased out as Specware is able to duplicate and supercede its funcationality.

A collection of programming theories can be organized into a refinement hierarchy using theory morphisms as the refinement arrow [9]. The question emerges of how to access and apply knowledge in such a hierarchy. The answer is illustrated in the "ladder construction" diagram on the left:





The left-hand side of the ladder is a path in the refinement hierarchy of algorithm theories starting at the root (Problem Theory). Oil_Flow_0 is a given specification theory of a problem. The ladder is constructed a rung at a time from the top down. The initial arrow (theory morphism) from problem theory to Oil_Flow_0 is trivial. Subsequent rungs are constructed abstractly as in the diagram on the right above, where $P_{i+1} \oplus S_i$ is the pushout theory and S_{i+1} is an extension of S_i determined by constructing the theory morphism m''_{i+1} (Techniques for constructing specification morphisms are presented in [6]). The morphism m_{i+1} is determined by composition.

Our classification approach to design is based on a hierarchic classification of design knowledge applied via the ladder construction. The goal is to find the strongest possible classification (or view) of the given problem by incrementally constructing morphisms. Morphisms from deeper theories in the hierarchy expose more structure in the given problem, thus enabling the synthesis of better algorithms.

I worked out a detailed example of the Ladder Construction in which the problem was to find an optimal flow of oil through a network of depots. The construction grounds out in the synthesis of interface code that invokes a fast FORTRAN program for solving network flow problems.

2. *Applications to Scheduling Problems* –

The U.S. Transportation Command and the component service commands use a relational database scheme called a TPFDD (Time-Phased Force and Deployment Data) for specifying the transportation requirements of an operation, such as Desert Storm or the Somalia relief effort. We developed a domain theory of TPFDD scheduling defining the concepts of this problem and developed laws for reasoning about them. KIDS (Kestrel Interactive Development System) was used to derive and optimize a variety of global search scheduling algorithms that perform constraint propagation [5, 10]. The resulting code, generically called KTS (Kestrel Transportation Scheduler), has been run on a variety of TPFDDs generated by planners at USTRANSCOM and other sites. With one such TPFDD problem, KTS was able to schedule 15,460 individual movement requirements in 71 cpu seconds. The schedule used relatively few resources and satisfied all specified constraints. KTS is orders of magnitude faster than any other TPFDD scheduler known to us.

We spent much of this year exploring techniques for handling various classes of resources. For example, certain resources have the property that they are asynchronously sharable but bounded; e.g. parking lots. The general techniques for modeling this property are highly reusable and thus worthy of representation. Using our formal method for deriving constraint propagation code (see next section), we derived a mechanism for handling such asynchronously sharable resources in the presence of time windows. Discussing this result with other researchers has suggested that this is a new result. Interestingly, we were unable to intuit the nature of this mechanism, but were able to use the formalism to calculate it (on paper).

We used these explorations of the common properties of resources in deriving a family of transportation schedulers, for increasingly rich models of the transportation domain. The results on asynchronously sharable resources is directly applicable to the handling of MOG (Maximum On Ground) constraints at ports, which deal with bounds on the available parking space.

Much of our work has been driven by attempting to derive scheduling codes that could be delivered and used at USTC (U.S. Transportation Command), AMC (Airlift Mobility Command), and PACAF (Pacific Air Command).

Theater Transportation Scheduling

The PACAF (Pacific Air Force) Airlift Operations Center at Hickam AFB, Honolulu is tasked with in-theater scheduling of a fleet of 26 C-130 aircraft (plus assorted strategic aircraft on loan) throughout the Pacific region. Current scheduling practice is essentially manual; for example, the relief effort for Hurricane Iniki which struck the island of Kauai in September 1992 was sketched out on 2 sheets of legal paper and required hours of labor. Since Spring 1994 researchers from Kestrel Institute and BBN, Cambridge have been working with personnel from PACAF to model the in-theater scheduling problem. The resulting domain theory has been used to synthesize an increasingly rich series of schedulers generically called ITAS (In-Theater Airlift Scheduler). ITAS runs on a laptop computer (Macintosh Powerbook) which makes it useful for both field and command center operations. BBN has built the user interface based on the commercial Foxpro database package. ITAS schedules the Hurricane Iniki data in a few seconds.

To produce "flyable" schedules it has been necessary to model and schedule a variety of resources, including aircraft, air crews and their duty days, ground crews, parking space for aircraft, and other port restrictions.

An alpha release of a scheduler running on an Apple Powerbook was delivered to PACAF at Hickam AFB, Honolulu in August 1994. This may be the first example of a machine-synthesized algorithm being delivered to a customer. Subsequent versions of ITAS have been used in several exercises and ITAS was the sole scheduler used in an international exercise during September 1995 (JWID-95). *ITAS is regarded as being ready to use for contingency operations by PACAF personnel.*

We have gone through many cycles of learning about the problem from the customer/end-user, elaborating our domain theory, generating new code, and observing PACAF personnel using the scheduler. Although this is a time-consuming process, it seems essential to developing an application that will be used. Nevertheless there has been significant payoff to us as researchers, since the problem features required by the end-user has forced us to generalize and deepen our theories of algorithm design.

Power Plant Outage Scheduling

We are continuing to develop new scheduling applications using KIDS. A joint project with the Electric Power Research Institute in Palo Alto, California and Rome Laboratory, focuses on the scheduling of maintenance activities during an outage period at nuclear power plants. KIDS is being used to model the problem and to generate high-performance schedulers for maintenance activities. Current schedulers used by the utility industry are slow and handle only a small subset of the important features of the problem. Safety constraints are extremely important, as well as the efficiency of the schedule, since an outage period can costs millions of dollars per day.

3. *Synthesis of Constraint Propagation Code*

In Constraint Programming, a constraint set partially characterizes objects of interest and their relationships. Constraint propagation is one of the key operations on constraints in Constraint Programming. As committments are made that further characterize some object, we want to infer consequences of those committments and add those consequences as new constraints. Efficiency concerns drive us to look closely at (1) the representation of constraints, (2) inference procedures for solving constraints and deriving consequences, and (3) the capture of inferred consequences as new constraints.

We have been studying constraint propagation in the context of global search algorithms. We have found a precise and abstract characterization of constraint propagation and a means for mechanically generating propagation code [12, 7, 11]. Propagation is essentially the iterative application of "cutting constraints" which are necessary conditions that every element of a set of candidate solutions is feasible (or optimal). Propagation can also be characterized as a specialized forward inference procedure. This abstract characterization allows us generalize and unify several special cases that have appeared in the literature: (1) Gomory cutting plane technology from the O.R. literature, and (2) constraint propagation methods in the CSP literature. We also believe that many iterative procedures used in scientific and numerical computing are special cases.

4. Software Evolution

   My colleague Y.V. Srinivas and I have begun to explore a formal approach to evolution of formal descriptions that is based on inference and dependency analysis. We view evolution as the transition from one consistent description to another. Each such transition can be decomposed into three phases: (1) start with a consistent description, (2) change some aspect of the description (possibly introducing inconsistency), (3) minimally change other parts of the description to re-establish consistency (change propagation). In general, change propagation is the maintenance of certain properties (such as consistency, well-formedness, etc.) while changing others.

   To make the problem of change propagation tractable, we restrict our attention to changes which are monotonic, i.e., generalizations or specializations (other changes can be represented as combinations of these). Using an explicit representation of a consistency property as a formula, we use a special form of inference, *directed inference*, to determine which parts of the description to change in order to re-establish the desired consistency property. The inference makes use of dependency information which indicates the direction and amount of change, *variance*, of each entity in the domain with respect to changes in other entities.

References

[1] LOWRY, M. R. *Algorithm Synthesis Through Problem Reformulation.* PhD thesis, Computer Science Department, Stanford University, 1989.

[2] SMITH, D. R. Top-down synthesis of divide-and-conquer algorithms. *Artificial Intelligence 27*, 1 (September 1985), 43–96. (Reprinted in *Readings in Artificial Intelligence and Software Engineering*, C. Rich and R. Waters, Eds., Los Altos, CA, Morgan Kaufmann, 1986.).

[3] SMITH, D. R. Structure and design of global search algorithms. Tech. Rep. KES.U.87.12, Kestrel Institute, November 1987.

[4] SMITH, D. R. Structure and design of problem reduction generators. In *Constructing Programs from Specifications*, B. Möller, Ed. North-Holland, Amsterdam, 1991, pp. 91–124.

[5] SMITH, D. R. Transformational approach to scheduling. Tech. Rep. KES.U.92.2, Kestrel Institute, November 1992.

[6] SMITH, D. R. Constructing specification morphisms. *Journal of Symbolic Computation, Special Issue on Automatic Programming 15*, 5-6 (May-June 1993), 571–606.

[7] SMITH, D. R. Toward the synthesis of constraint propagation algorithms. In *Logic Program Synthesis and Transformation (LOPSTR '93)*, Y. DeVille, Ed. Springer-Verlag, 1993, pp. 1–9.

[8] SMITH, D. R. KIDS – a semi-automatic program development system. *IEEE Transactions on Software Engineering Special Issue on Formal Methods in Software Engineering 16*, 9 (September 1990), 1024–1043.

[9] SMITH, D. R., AND LOWRY, M. R. Algorithm theories and design tactics. *Science of Computer Programming 14*, 2-3 (October 1990), 305–321.

[10] SMITH, D. R., AND PARRA, E. A. Transformational approach to transportation scheduling. In *Proceedings of the Eighth Knowledge-Based Software Engineering Conference* (Chicago, IL, September 1993), pp. 60–68.

[11] SMITH, D. R., PARRA, E. A., AND WESTFOLD, S. J. Synthesis of high-performance transportation schedulers. Tech. Rep. KES.U.95.6, Kestrel Institute, March 1995.

[12] SMITH, D. R., AND WESTFOLD, S. J. Synthesis of constraint algorithms. In *Principles and Practice of Constraint Programming*, V. Saraswat and P. V. Hentenryck, Eds. The MIT Press, Cambridge, MA, 1995.

P.I. Name: Douglas R. Smith
Insitution: Kestrel Institute
Telephone: (415) 493-6871
E-mail: smith@kestrel.edu
Contract Title: Foundations of Software Development
Contract Number: N00014-93-C-0056
Reporting Period: 1 Oct 92 - 30 Sep 95

3. Lists of Publications, Presentations, and Reports

3.1. Publications

1. Smith, D.R., Derivation of Parallel Sorting Algorithms, in *Parallel Algorithm Derivation and Program Transformation*, Eds. R. Paige, J. Reif, and R. Wachter, Kluwer Academic Publishers, 1993, 55-69.

2. Smith, D.R., Automating the Design of Algorithms, in *Formal Program Development, IFIP TC2/WG2.1 State-of-the-Art Report*, Eds. B. Möller, H. Partsch, S. Schumann, LNCS 755, Springer-Verlag, 1993, 324-354.

3. Smith, D.R., Constructing Specification Morphisms, Journal of Symbolic Computation, Special Issue on Automatic Programming, Vol 16, No 5-6, 1993, 571-606.

4. Smith, D.R., Synthesis of Constraint Algorithms, in *Proceedings of Principles and Practice of Constraint Programming (PPCP93)*, Newport, RI, April 1993.

5. Smith, D.R. and Parra, E.A., Transformational Approach to Transportation Scheduling, in *Proceedings of the Eighth Knowledge-Based Software Engineering Conference*, (Best Paper Award), IEEE Computer Society Press, September 1993, 60-68.

6. Smith, D.R., Synthesis of Constraint Algorithms, in *Proceedings of Third International Workshop on Logic Program Synthesis and Transformation (LOPSTR '93)*, Louvain-la-Neuve, Belgium, July 1993.

7. Smith, D.R., Toward the Synthesis of Constraint Propagation Algorithms, in *Logic Program Synthesis and Transformation*, Y. Deville (Ed.), Workshops in Computing Series, Springer-Verlag, 1994, 1-9.

8. Smith, D.R. and Parra, E.A., Transformational Approach to Transportation Scheduling, in "Proceedings of the ARPA/Rome Lab Planning Initiative Workshop", Tucson, Arizona, February, 1994, 205-216.

9. Smith, D.R. and Westfold, S.J., Synthesis of Constraint Algorithms, in "Principles and Practice of Constraint Programming", V. Saraswat and P. Van Hentenryck (Eds.), MIT Press, 1995, 173-182.

10. Smith, D.R., Application of Program Synthesis Technology, Sidebar in Special Issue on the Rome/ARPA Planning Initiative, IEEE Expert 10(1), February 1995, 11.

11. Smith, D.R., Toward Practical Applications of Software Synthesis, in Proceedings of the Workshop on Formal Methods in Software Engineering, ICSE-17, Seattle WA, May 1995.

12. Smith, D.R. and Green, C.C., Toward Practical Applications of Software Synthesis, to appear in Proceedings of the Formal Methods in Software Practice Workshop, San Diego, CA, January 1996.

13. Gomes, Carla O.P. and Smith, D.R., An Integrated Approach Towards Planning and Scheduling – An Application to Outages of Power Plants, submitted to AIPS-96.

14. Burstein, M.B. and Smith, D.R., ITAS: A Portable Interactive Transportation Scheduling Tool Using a Search Engine Generated from Formal Specifications, submitted to AIPS-96.

15. Pepper, P. and Smith, D.R., A High-level Derivation of Global Search Algorithms (with Constraint Propagation), submitted to Science of Computer Programming, Special Issue on Formal Methods, 1995.

3.2. Presentations

Douglas R. Smith

1. Presented a talk and demo of KIDS, "Recent Progress in Program Synthesis", Xerox PARC Forum, Thursday, 18 March 1993.

2. All-day seminar at Boeing Computer Services, Formal Methods Lecture Series, Bellevue, Wash ington, 29 March 1993 (three lectures and two extended demos of KIDS).

3. CS409 Automated Algorithm Design, Stanford University, April-June 1993.

4. Presented talk on "Synthesis of Scheduling Algorithms" and KIDS demo, BBN, Cambridge, MA, April 1993.

5. Presented talk on "Synthesis of Constraint Algorithms", Principles and Practice of Constraint Programming (PPCP93), April 1993.

6. Presented talk on "Synthesis of Scheduling Algorithms", IFIP Working Group 2.1 meeting, Winnipeg, Manitoba, Canada, 10-14 May 1993.

7. Presented talk on "Synthesis of Scheduling Algorithms" and KIDS demo, Rome Laboratories, Rome, NY, 20 May 1993.

8. Presented talk "Classification Approach to Algorithm Design" and KIDS demo, Imperial College, London, 6 July 1993.

9. Presented invited talk on "Synthesis of Constraint Algorithms" and KIDS demos, LOPSTR '93, Louvain-la-Neuve, Belgium, 7-9 July 1993.

10. Presented talk "Classification Approach to Algorithm Design" and KIDS demo, Technical University of Berlin, 12 July 1993.

11. Presented a talk and demo of KIDS, "Recent Progress in Program Synthesis", GMD, Aldershof, Germany, 13 July 1993.

12. Presented talk "Classification Approach to Algorithm Design" and KIDS demo, University of Ulm, Germany, 14 July 1993.

13. Presented a talk and demo of KIDS, "Recent Progress in Program Synthesis", Siemens, Munich, Germany, 15 July 1993.

14. All-day Tutorial on "KIDS: An Algorithm Synthesis System", Eighth Knowledge-Based Software Engineering Conference, McLean, VA, 20 September 1993.

15. Presented a talk, "Transformational Approach to Transportation Scheduling", and KIDS demo, Eighth Knowledge-Based Software Engineering Conference, McLean, VA, 20-23 September 1993.

16. Presented a talk on "Automated Software Development" and KTS demo, UNISYS Corporation, 6 Oct 93, McLean, VA.

17. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, ARPA/RL Air Campaign Planning Workshop, 8 Oct 93, Arlington, VA.

18. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, US Transportation Command (USTRANSCOM), Scott AFB, IL, 13 Oct 93.

19. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, Air Mobility Command (AMC), Scott AFB, IL, 14 Oct 93.

20. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, McGuire AFB, NJ, 9 Nov 93.

21. Presented a tutorial talk on KIDS plus KIDS demo, Naval Surface Warfare Center (NSWC), White Oak, MD, 11-12 Nov 93.

22. Taught KIDS workshop (with Major P. Bailor (AFIT)), AFIT, Wright-Patterson AFB, Ohio, 13-17 December 1993.

23. Chaired IFIP WG2.1 meeting, and presented talk on "Classification Approach to Design", Renkum, The Netherlands, 10-13 Jan 94.

24. Invited presentation, AI/OR workshop, Rome Laboratory, 27-28 Jan 94.

25. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, ARPA/Rome Lab Planning Initiative Workshop, Tucson, Arizona, 22-24 February, 1994.

26. Presented a talk on "Automated Software Development" and KIDS demo, Workshop on Logical Theory for Program Construction, Schloss Dagstuhl, Germany, 7-11 March 1994.

27. Presented a talk on "Automated Software Development" and KTS demo, Apple Computer, Cupertino, CA, 14 Mar 94.

28. Presented a talk on "Automated Software Development" and KTS demo, Sun Microsystems, Mtn View, CA, 15 Mar 94.

29. Presented a KTS demo, BBN, Cambridge, MA, 15 Apr 94.

30. Presented talk on "Transformational Approach to Transportation Scheduling" and KIDS demo, ARPA Software Engineering Foundations workshop, Herndon, VA, 16-17 Jun 94.

31. Taught 5-day KIDS workshop (with Major P. Bailor (AFIT) and Y.V. Srinivas (Kestrel)), NSA, Fort Meade, MD, 8-12 Aug 94.

32. Invited talk on "Automated Software Development" and KIDS demo, HP Software Technology Lecture Series, Palo Alto, CA, 1 Sept 94.

33. Presented a talk on "Transformational Approach to Transportation Scheduling", AFOSR Contractors meeting, Bolling AFB, Washington, DC, 12-13 September 1994.

34. Presentation and KTS demo, ARPA, Arlington VA, 14 Sept 94.

35. Taught a tutorial on "KIDS: An Algorithm Synthesis System", Ninth Knowledge-Based Software Engineering Conference, Monterey, CA, 20 September 1994.

36. Chaired IFIP WG2.1 meeting, and presented talk on "Synthesis of Constraint Propagation Algorithms", UHK, Hong Kong, 9–13 Jan 95.

37. Invited talk on Refinement Approach to Parallel Software Engineering, Rome Laboratory Parallel Forecast Engineering Software Panel, 24-25 January 1995, Orlando, FL

38. Lectures and Demo, KIDS Workshop, AFIT, WPAFB, Dayton, OH, 24 Mar 95.

39. Presented paper on "Practical Applications of Software Synthesis", Workshop on Formal Methods Application in Software Engineering Practice, 17th International Conference on Software Engineering, Seattle, Washington, 24-25 April 1995.

40. KIDS/KTS/ITAS demos, ARPA SSTO Symposium, Chantilly, Virginia, 29-31 August, 1995.

3.3. Technical Reports

1. Smith, D.R., Transformational Approach to Scheduling, Technical Report KES.U.92.2, Kestrel Institute, Palo Alto, CA, November 1992, 54 pages.

2. Smith, D.R., Classification Approach to Design, Technical Report KES.U.93.4, Kestrel Institute, Palo Alto, CA, November 1993, 24 pages

3. Douglas R. Smith, Eduardo A. Parra, Stephen J. Westfold, Synthesis of High-Performance Transportation Schedulers, Report KES.U.95.6, Kestrel Institute, Palo Alto, CA, 1995.

P.I. Name: Douglas R. Smith
Insitution: Kestrel Institute
Telephone: (415) 493-6871
E-mail: smith@kestrel.edu
Contract Title: Foundations of Software Development
Contract Number: N00014-93-C-0056
Reporting Period: 1 Oct 92 - 30 Sep 95

4. Description of Research Transitions and DoD Interactions

The main "transition" of the ONR-sponsored work has been through our experimental development system, KIDS. We have received many requests for the system from researchers in software automation. Copies of KIDS are now installed at over 40 sites including Air Force Institute of Technology, Wright-Patterson AFB (Bailor and students), Catholic University of Louvain, Belgium (Sintzoff, Ledru), Technishe Hochschule Darmstadt, Germany (Bibel, Kreitz), Naval Postgraduate School, Monterey (Luqi), Andersen Consulting, Chicago (DeBellis, Miralya), Imperial College, London (Maibaum), and Information Sciences Institute, USC (Balzer, Feather).

Dr. Smith made several presentations to DoD and Government personnel during the contract period:

1. Dr. Larry Hatch from NSA spent two weeks at Kestrel working closely with Dr. Smith during November 1992 developing a new algorithm.
2. Dr. Smith presented a talk on "Synthesis of Scheduling Algorithms" and KIDS demo, Rome Laboratories, Rome, NY, 20 May 1993.
3. In August 1994 we delivered an alpha release of a in-theater scheduler to PACAF at Hickham AFB, Hawaii. During 26-29 Sept 1994 we will be installing a beta release. The system, called ITAS (In-Theater Airlift Scheduler), is a joint development of BBN and Kestrel and runs on an Apple Powerbook. The interface to ITAS is being developed by BBN and the scheduler is being synthesized using KIDS by Kestrel personnel.
4. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, ARPA/RL Air Campaign Planning Workshop, 8 Oct 93, Arlington, VA.
5. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, US Transportation Command (USTRANSCOM), Scott AFB, IL, 13 Oct 93.
6. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, Air Mobility Command (AMC), Scott AFB, IL, 14 Oct 93.
7. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, McGuire AFB, NJ, 9 Nov 93.
8. Presented a tutorial talk on KIDS plus KIDS demo, Naval Surface Warfare Center (NSWC), White Oak, MD, 11-12 Nov 93.
9. Taught KIDS workshop (with Major P. Bailor (AFIT)), AFIT, Wright-Patterson AFB, Ohio, 13-17 December 1993.

10. Presented a talk on "Transformational Approach to Transportation Scheduling" and KTS demo, ARPA/Rome Lab Planning Initiative Workshop, Tucson, Arizona, 22-24 February, 1994.

11. Presented talk on "Transformational Approach to Transportation Scheduling" and KIDS demo, ARPA Software Engineering Foundations workshop, Herndon, VA, 16-17 Jun 94.

12. Taught 5-day KIDS workshop (with Major P. Bailor (AFIT) and Y.V. Srinivas (Kestrel)), NSA, Fort Meade, MD, 8-12 Aug 94.

13. Presented a talk on "Transformational Approach to Transportation Scheduling", AFOSR Contractors meeting, Bolling AFB, Washington, DC, 12-13 September 1994.

14. Presentation and KTS demo, ARPA, Arlington VA, 14 Sept 94.

15. The ITAS scheduler that we synthesized (collaborating with BBN) has been delivered to PACAF Airlift Operations Center, Hickham AFB and extensively tested in exercises. In particular, ITAS was the sole scheduler used in the multi-national JWID-95 (Joint Warrior Integrated Demonstration) exercise during September 1995. ITAS is regarded by PACAF/AOM as being ready to use for contingency purposes.

16. On-site interactions with personnel with PACAF Airlift Operations Center, Hickham AFB, HI during 14-16 Feb 1995. Synthesized and installed an improved version of ITAS.

17. ITAS Demo, C2IPS meeting (an Air Force base-level command and control system under construction), Computer Sciences Corp, Moorestown, NJ, 23 March 1995.

18. Taught KIDS workshop (with Major P. Bailor (AFIT)), AFIT, Wright-Patterson AFB, Ohio, 24 March 1995.

19. Presented KIDS/KTS/ITAS demos at the ARPA SSTO Symposium, Chantilly, Virginia, 29-31 August, 1995.

20. Meeting with personnel from the Checkmate office at the Pentagon about air campaign scheduling systems, 13-14 September 1995.

P.I. Name: Douglas R. Smith
Insitution: Kestrel Institute
Telephone: (415) 493-6871
E-mail: smith@kestrel.edu
Contract Title: Foundations of Software Development
Contract Number: N00014-93-C-0056
Reporting Period: 1 Oct 92 - 30 Sep 95

5. Description of Software and Hardware Prototypes

(1) KIDS – The Kestrel Interactive Development System (KIDS) provides an open architecture for experimenting with the semi-automated development of formal specifications into correct and efficient programs. The system has components for performing algorithm design, deductive inference, program simplification, partial evaluation, finite differencing optimizations, data type refinement and other development operations. Although their application is interactive, all of the KIDS operations are automatic except the algorithm design tactics which require some interaction at present. Over sixty programs have been derived using the system and we believe that KIDS could be developed to the point that it becomes economical to use for routine programming. We are not currently working on commercializing this system – it is regarded purely as an experimental testbed.

(2) Specware – Current and pending projects at Kestrel focus on the development of a theory-based system called SPECWARE that succeeds our previous research prototypes (KIDS, REACTO, DTRE). SPECWARE is designed to be a robust, open-architecture, well-documented, easy-to-use software developement system. SPECWARE aims to integrate the algorithm design capabilities of KIDS and the data type refinement capabilities of DTRE on a unified formal basis. An important additional goal is to scale up from algorithm design to system design. The development of SPECWARE is structured to address, in turn, the construction of a robust kernel (complete), a fully functional, usable system, and the creation of a very low cost, widely available version.

(3) KTS (Kestrel Transportation Scheduler) KTS is a strategic air/sea-lift TPFDD scheduler synthesized from a Refine specification using KIDS.

Inputs: TPFDD, situation model, geolocs
User interaction: editting TPFDD, situation model, geoloc database if necessary
Outputs: schedule
Implementation: Common Lisp/CLIM on Sun workstations and Macs. KTS can be run remotely through the WWW by accessing http://kestrel.edu/www/demos.html

(4) ITAS (In-Theater Airlift Scheduler) ITAS schedules in-theater cargo aircraft, their crews, ground unload crews and parking at ports (mog). ITAS was also synthesized from a Refine specification using KIDS.

Inputs: movement requirements, situation model, aircraft model
User interaction: editting the inputs if necessary, also interaction modification of the schedule.
Outputs: schedule (Gantt chart display)
Implementation: Common Lisp and MS Foxpro on Macs.