Miguel Filgueiras Luís Damas (Eds.)

Cc01-727

Progress in Artificial Intelligence

6th Portuguese Conference on AI, EPIA '93
Porto, Portugal, October 6-8, 1993
Proceedings

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editor

Jörg Siekmann
University of Saarland
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3,
D-66123 Saarbrücken, Germany

Volume Editors

Miguel Filgueiras
Luís Damas
LIACC, Universidade do Porto
R. do Campo Alegre 823, 4100 Porto, Portugal

6385

CR Subject Classification (1991): I.2.3-6, I.2.7, I.2.11

ISBN 3-540-57287-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57287-2 Springer-Verlag New York Berlin Heidelberg

Typesetting: Camera ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

These are the proceedings of the 6th Portuguese Conference on Artificial Intelligence (EPIA'93) organised by the Portuguese Artificial Intelligence Association. Like the last two conferences in this series, this one was run as an international event with strict requirements as to the quality of accepted submissions. Fifty one submissions were received from 9 countries, the largest numbers coming from Portugal (18), Germany (10), and France (8).

With a few exceptions, submissions were evaluated by three referees, who were asked to comment on any reports where the overall evaluations did not agree. At the Programme Committee meeting, these cases as well as those in which only one or two reports were available were carefully examined. The Programme Committee decided on the acceptance of 25 out of the original 51 submissions. A further 8 were selected as posters, 7 of which have their abstracts included here. The members of the Programme Committee and the referees are listed below.

We had the honour to have as invited lecturers David H. D. Warren, Les Gasser, and Yoav Shoham, their presentations highly contributing to the interest and quality of the conference. Les Gasser prepared a written summary of his lecture for this volume. To all three our sincere thanks.

Our thanks also extend to all those who contributed in any form to make this conference possible namely, the Programme Committee members, the referees, the authors, the other members of the Organising Committee and the following institutions (in alphabetical order): Câmara Municipal do Porto, Centro de Informática da Universidade do Porto, Commission of the European Communities, Digital Equipment Portugal, Fundação António Almeida, Fundação Calouste Gulbenkian, Fundação Luso-Americana para o Desenvolvimento, IBM Portuguesa, and Junta Nacional de Investigação Científica e Tecnológica.

Porto, June 1993

Miguel Filgueiras
Luís Damas

Programme Chairmen

Miguel Filgueiras, Luís Damas

Programme Committee

António Porto, Universidade Nova de Lisboa
Armando Matos, Universidade do Porto
David Warren, University of Bristol
Ernesto Costa, Universidade de Coimbra
Eugénio Oliveira, Universidade do Porto
Jean-Louis Lassez, T. J. Watson Research Center, IBM
Luís Moniz Pereira, Universidade Nova de Lisboa

Organising Committee

Miguel Filgueiras, Nelma Moreira, Rogério Reis, Ana Paula Tomás

Referees

Alípio Jorge	Amílcar Cardoso	Ana Paula Tomás
António Mário Florido	António Porto	Armando Matos
Artur Miguel Dias	Carlos Bento	Claire Willis
Cristina Ribeiro	David H. D. Warren	Ernesto Costa
Eugénio Oliveira	Fernando Moura Pires	Fernando Silva
Francisco Menezes	Gabriel David	Joaquim Baptista
Joaquim Correia	John Galagher	John Lloyd
José C. Cunha	José Ferreira	José Júlio Alferes
José Paulo Leal	Giovanni Varile	Luís Caires
Luís Damas	Luís Moniz Pereira	Luís Monteiro
Luís Torgo	Margarida Mamede	Maria Benedita Malheiro
Michael Fisher	Michael Wooldridge	Michel Wermelinger
Miguel Filgueiras	Nelma Moreira	Nuno Mamede
Paulo Moura	Rogério Reis	Sabine Broda
Steve Gregory	Anonymous	

Table of Contents

Organizations as Complex, Dynamic Design Problems

Les Gasser, Ingemar Hulthage, Brian Leverich,
Jon Lieb, and Ann Majchrzak
Computational Organization Design Lab
Institute of Safety and Systems Management
USC, Los Angeles, CA. 90089-0021 USA
(213) 740-8771
{<lastname> | majchrza}@usc.edu

Abstract. The ACTION organization design and analysis system is a research and development effort designed to assist business re-engineering and organizational or technology change by helping to improve the integration of technology, organizations, and people ("TOP-integration") in manufacturing enterprises. ACTION uses a multi-level constraint-based representation of organizational features including business objectives, unit structure, skills needed, performance monitoring/reward systems, decisionmaking discretion, employee values, coordination attributes, etc. to both evaluate existing organization designs and to help users develop new ones. ACTION's core software is domain-independent, theory-driven architecture designed for application to a wide range of design and analysis problems.

1 Introduction

The ACTION organization design and analysis system is a research and development effort designed to assist business re-engineering and organizational or technology change by helping to improve the integration of technology, organizations, and people ("TOP-integration") in manufacturing enterprises. ACTION helps analyze interactions among a wide range of technological and organizational features, and to design flexible and adaptive organizations that optimize a range of business-oriented objectives. The goal is to assure that the use of ACTION as part of a change management process will lead to

- Improved assessment of how/where organizations or technologies need to be changed to meet specific business objectives
- Increased confidence that fewer technology and business-process implementation issues are overlooked during planning
- More accurate assessment of technology plans in concurrent engineering processes
- Greater assurances that technology modernization plans will succeed
- Identifying the organizational and human costs of capital investment and business strategy decisions

- Increased awareness of alternatives through prior what if analyses
- Increased awareness of design options generated by ACTION
- Initiating discussions with technology planners and users
- Increased awareness of workforce (re)skilling and (re)training opportunities and needs.

1.1 Technology-Organization-People Integration

The ACTION project views TOP-integration as a problem of *matching and coordinating* numerous technological and organizational features (rather than optimizing just one of them), and of *optimizing this match to fit organizational business objectives and environmental circumstances*. ACTION's theory, methodology, and software efforts incorporate knowledge of technological and organizational features including

- Organizational objectives
- Values held by employees
- Performance management systems
- Customer involvement
- Decision-making discretion
- Skills and training needs and opportunities
- Information, tool, and technology resources
- Coordination among jobs and units
- Organizational unit structure
- Job design
- Variances (e.g., turnover, materials quality)
- Specific technical system characteristics

The core knowledge developed in the ACTION Project---a comprehensive model of positive and negative relationships among these key organizational features---is called the ACTION TOP-integration Theory. This theory is a general, domain-independent theoretical and conceptual framework.

The theory captures and represents a large number of manufacturing TOP features that trade-off with one another, in an *open-systems* (OS) model. The open-systems model is a structure that allows for dynamically matching and coordinating these features (rather than optimizing just one of them), and optimizing this match to fit a collection of selected objectives (see below) and circumstantial variances (such as materials quality, process variances, personnel turnover, attendance, etc.) The OS model treats each of these TOP factors as a variable in a large network of constraints. The OS theory specifies the variables, their potential values, and the mutually reinforcing or constraining relationships among them. The purpose of this model is to allow the designer/analyst 1) to capture and track a very large number of TOP features and their interactions, 2) to discover specific points of congruence and misalignment among features, as a basis for redesign or trade-off analysis for optimizing the environmental conservation objective, and 3) to flexibly revise and explore alternative model inputs and outputs and their impacts. ACTION also includes a set of domain-specific refinements to this framework that specialize ACTION's TOP-integration model to production areas of discrete-parts manufacturing organizations, and within this category, to four specific types of production context: short-life-cycle and general group technology cells, functional shops, and transfer lines.

The theory and tool have been developed as follows:

- Critical concepts and general statements of relationships were identified via a qualitative meta-analysis of empirical studies of successful and unsuccessful TOP integration in discrete parts manufacturing.
- To specify specific relationships for specific variables, a knowledge elicitation process was designed for use with industry TOP integration experts. The process included delphi, consensus-building, stable membership (over 1.5 years) in a technical development team, selection of team members to represent a wide diversity of industry contexts, explicit documentation and agreement on all terminology, use of matrices and decision trees to specify relationships, and iterations on theory and applications.
- The elicitation process proceeded over 1.5 years with team members from Digital Equipment Corporation, General Motors, Hewlett Packard, Hughes Aircraft Company, and Texas Instruments.
- The ACTION software decision support and analysis tool, designed for easy theory revision and enhancement, was developed with continuous industry review and input.
- The theory, structure, and usability features of ACTION are now being validated through pilot tests and via a study that incorporates detailed data from 100 sites.

ACTION supports four key functionalities:

- Summary-level analysis and design of TOP integration
- Detailed-level critique and evaluation of TOP integration
- Detailed-level design of TOP-integration
- Explanation of analysis/design concepts and results

ACTION represents the TOP-integration features in a design or evaluation problem as a set of business objectives (goals) to be optimized and a set of detailed, hierarchical constraints (called the theory minimodels) that describe how organizational features, such as above, interact to affect the level of optimization achieved for these objectives. Conversely, the mini-models describe how a required level of optimization, implies constraints on organizational features. ACTION's knowledgebase contains detailed constraint models of the possibility of achieving seven organizational objectives:

- Minimizing throughput time
- Maximizing product quality
- Maximizing employee flexibility
- Maximizing product responsiveness
- Maximizing process responsiveness
- Maximizing changeover responsiveness
- Maximizing manufacturability of designs

To guide the process of generating evaluations or design solutions, a user employs ACTION to select and manipulate constraints, trade-offs, and value-assignments for organization and technology variables that form the theory minimodels.

The ACTION project has built upon the knowledge and insights gained from two prior efforts: the HITOP (Highly-Integrated Technology, Organizations, and People) approach to analyzing human infrastructure impacts of advanced manufacturing technologies [Majchrzak 88; Majchrzak et al. 91], and the HITOP-A (HITOP-Automated) decision support system [Gasser and Majchrzak 92, Majchrzak and Gasser 92]. The HITOP-A system was initially developed at USC. Using decision rules and heuristics, this system predicted human infrastructure required for a particular state of organization, job design, and technology variables. The focus of HITOP-A was predictive decisionmaking and support for flexible manufacturing cells. Work on HITOP-A, completed in 1991, produced an operational prototype which was capable of developing the job and organizational designs that incorporate organizational and technological knowledge and input.

2 Overall System Strategies

The knowledge-based system software core of ACTION is a general, domain-independent tool that operationalizes ACTION domain-specific TOP theories or other theories, to create a combined analysis, design, and explanation/teaching system for a specific domain. This general modeling framework has been used to implement a software decision-support system that includes the ACTION discrete-parts manufacturing models. The overall strategy for ACTION has been to develop a core set of representation and knowledge structures which can be used as a single underlying theory and framework for accomplishing all of the key functionalities. These core knowledge structures are flexible and reconfigurable, so that they can be modified and improved without major software rebuilding. This unified approach leads to greater consistency, coherence, and integration across functionalities in the ACTION system, and improves maintainability. The core ACTION software is reconfigurable to incorporate new domain, control, or interface approaches. This means that the basic structure of the ACTION software and theory is directly applicable to a range of other problems (e.g. enterprise-level modeling), by "plugging in" new domain theories, design processes, and interface models.

This set of knowledge structures includes representations of

- Core ACTION domain theory (minimodels, theory matrices, business objectives/goal, user constraints, etc.)
- User interaction, exploration, explanation, and information facilities
- A constraint based reasoning system.

- Control knowledge such as design and analysis heuristics (e.g. priority rankings of goals or of minimodel elements), or system knowledge states (ACTION's record of what it has accomplished and what a user knows) and design process rules (ACTION's rules that capture recommendations about what ACTION steps to follow in what order, depending on a user's purpose).
- Theory abstractions such as matrices for summary-TOP analyses.

This set of representations is used simultaneously to achieve the four key ACTION functionalities. For example, design, evaluation, and explanation are all achieved by applying knowledge represented in the minimodels as constraints. For evaluation, a user specifies details of organizational unit structures and unit attributes, and the minimodels are applied unit-by-unit to derive information about what goals are achievable and what degree of match exists among an organizations attributes. For design, a user specifies a set of goals for an organization to meet, along with key design constraints, and ACTION uses the minimodels and unit design information to derive a unit structure and appropriate unit and technology attributes. For explanation, particular user-entered or system-derived values can be rationalized by reference to their place in the ACTION Domain Theory and prior minimodel-based conclusions.

ACTION allows for analysis and design of organizations at both summary and detailed levels. The strategy for accomplishing summary-level analyses that are coherent with detailed analyses is to use the same underlying knowledge and representation structures as a basis for both. Knowledge used for summary TOP analysis and design is simply abstracted from the knowledge used for detailed analysis and design. For example, the organizational goal **possibility of minimizing throughput** is linked to the detailed concept **percentage of technology that is reliable** through a dense network of concepts and relationships in the *throughput minimodel*. However, in the summary TOP analysis, this complex relationship has been simplified, and is represented as a pairwise constraint in the summary relationship matrix. In this way, summary relationships directly reflect underlying detailed theory, in abstracted form.

Summary TOP analyses focus on the relationships between organizational goals and key summary analytical variables. In addition, Summary TOP analysis provides high-level descriptions of basic unit design features, requirements, and strategies, taken from a small class of possibilities (rather than from a large class of detailed unit designs).

3 Data-Driven Approach to ACTION

Most routine knowledge-based systems development projects are based on capturing a well-understood (but possibly unarticulated) body of pre-existing knowledge. The ACTION development project has taken on an additional major challenge: the structure and the content of the theoretical and practical knowledge at the core of ACTION were not yet well understood at the outset of the project. Instead, these are under development concurrently with the structure and processing architecture of the ACTION software system. Thus ACTION is an exercise in a type of concurrent engineering, wherein a product (the theoretical and practical knowledge and methodology of ACTION) and a process (the set of tools, design and analysis algorithms, and usability features) are jointly and concurrently developed.

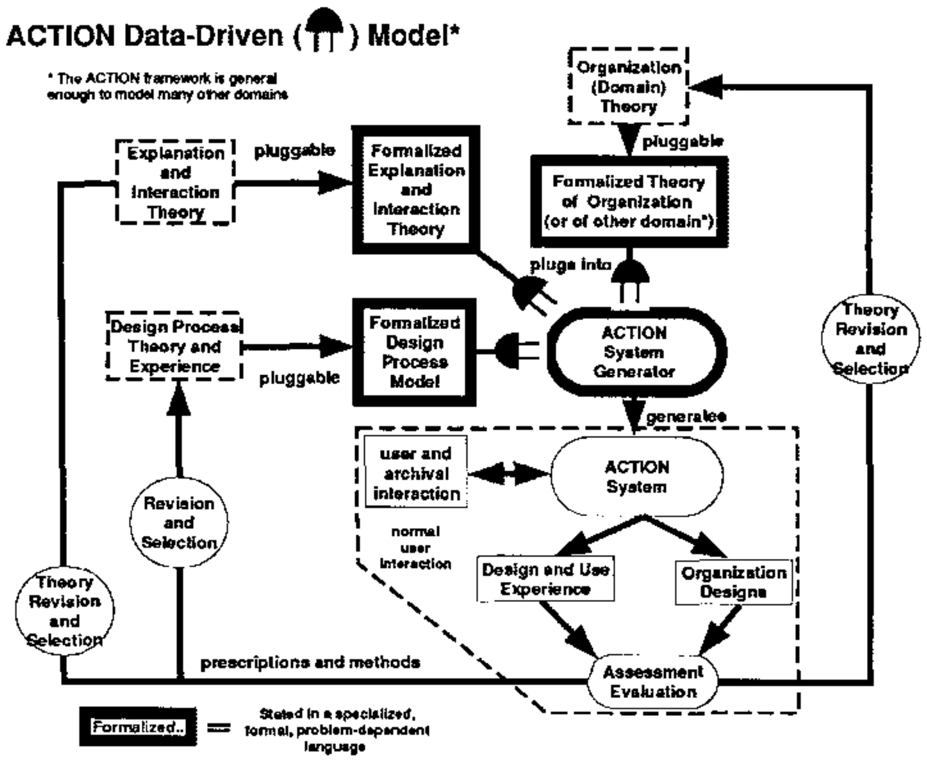
One challenge for ACTION development has been how to maintain the flexibility to accommodate revisions to both ACTION architecture and to ACTION knowledgebase content and structure throughout the development period and beyond, so that ACTION can be as responsive as possible to the needs of users and theory builders. One approach to such concurrent design is *design through abstraction*. Under this approach (which we are using), components of ACTION are being described, designed, and prototyped at successively finer levels of detail. Assumptions behind and interactions among components that arise along the way are refined iteratively as more and more knowledge becomes settled and codified, and more design commitments get made.

A complement to this iterative-abstraction-and-refinement approach to ACTION system development is the ability to remove and insert new component models and new functionalities as knowledge changes and as the level of detail increases. To do this, the ACTION system itself comprises a collection of replaceable and reconfigurable components. It is this replaceable and reconfigurable aspect that we term the *Data-Driven Model* approach to ACTION.

System architecture has been defined as a fixed, bounded, and static framework within which some variation in behavior and specification are possible [Erman 90]. Applying this viewpoint to ACTION, what needs to be defined is what parts of ACTION are static, fixed, and bounded, and what parts are flexible and allow for some variation in their behavior (see figure).

4 Organization Design

We treat organization design as a routine design problem (as vs. a creative design problem), with a well-defined space of possibilities and explicit evaluation criteria. Our approach is to model a designed artifact (in organization design this is an organization or *human infrastructure*) as a collection of variables with accompanying sets of possible values for each variable, at several interrelated levels of abstraction. These variables and values create a space of possible designs. For example, if we are designing chairs, we might use *strength-of-legs* as a design variable, with possible values being integers ranging from 10-100. A set of relations among these variables defines the hard evaluation criteria for designs, and divides the overall design space into acceptable designs (those in which the relations hold) and unacceptable designs (those in which the relations do not hold). The relations may be viewed as 1) constraints among variable values, 2) desired levels of correlation between variable values, or 3) desired levels of congruence (qualitative match) among variable values. For chairs, one such constraint might be *weight of chair + weight of heaviest*


ACTION Data-Driven ( ) Model*
* The ACTION framework is general enough to model many other domains
Organization (Domain) Theory
pluggable
Explanation and Interaction Theory
pluggable
Formalized Explanation and Interaction Theory
Formalized Theory of Organization (or of other domain*)
plugs into
Design Process Theory and Experience
pluggable
Formalized Design Process Model
ACTION System Generator
Theory Revision and Selection
generates
user and archival interaction
ACTION System
normal user interaction
Revision and Selection
Design and Use Experience
Organization Designs
Theory Revision and Selection
prescriptions and methods
Assessment Evaluation
Formalized.. = Stated in a specialized, formal, problem-dependent language

occupant must be less than or equal to strength of legs. Finally, within the space of acceptable designs, there is a set of evaluation criteria that describes "better" and "worse" designs.

In the abstract, then, a design problem is the problem of searching the space of possible designs (i.e., possible value assignments to design variables) for acceptable, highly-evaluated designs. Operators in this search include narrowing the size of sets of possible variable values, e.g. by assigning particular values to variables and propagating/analyzing the effects of the narrowed value sets. Heuristics include domain-specific search-ordering knowledge and domain-independent ordering analyses that depend on structural characteristics of the search space. The set of design description variables together with the set of relations, evaluation criteria and domain-dependent heuristics, is called the *organization design domain theory.* We include provisions for revisions to the domain theory, creating, in effect, a higher-order (theory-level) and more open search space. This allows manual or automatic searches through alternative domain theories, to find design spaces that encompass more appropriate designs.

4.1 Design Process Models

A design process model (DPM) is an explicit model of the activities carried out while creating a design, i.e., in searching a design space. Some of these activities and some aspects of the composition and structure of these activities are known before the design is undertaken, while other aspects of the design process emerge while creating the design, due to user inputs and the propagated effects of design choices. Thus, a design process model can be viewed as a control architecture with accompanying control knowledge for a design-problem solving system that (usually) involves people and automated design support tools. A DPM can also be viewed as a specification of both allocation of design activities (e.g. to people or to the support tools) and control choices (what to do and in what order to do it). Viewed from this perspective, a DPM needs to include the following elements:

- A set of control actions and heuristics, to establish strategies or goals for different modes of use.
- A set of actions that a design system can take in refining a design.
- A definition of user involvement, that is, which choices and activities are explicitly allocated to users and which to the automated parts of the system.
- A (partial) ordering of selected design activities that prescriptively describes the steps to be taken to generate and evaluate a design.
- A set of preferences and heuristics under which to make under-specified design choices that arise while generating a design.

Adopting a particular DPM amounts to making the hypothesis that, over a collection of design problems, a user using the design system will be able to generate a useful percentage of highly-evaluated acceptable designs with acceptable levels of effort.

4.2 Design as the Inverse of Evaluation

Evaluation criteria themselves can be seen as variables, each of which has an associated set of possible values; i.e., there is a space of possible design evaluations, independent of any particular design. Each of these variables can be seen as a characteristic of the design—that is, a way of describing a design in terms useful for evaluation. For example, a chair might be evaluated using a characteristic that directly reflects a design variable, such as *strength-of-legs*, or using a characteristic that relates to the design variables only indirectly, such as *cost*. The evaluation characteristics might have different possible values than those of the corresponding design variables, (e.g. high-med-low vs. 10-100). Evaluation criteria may also be joined using relations, such as an ideal strength-to-cost relation for chairs.

In general, then, we can think of an evaluation as a mapping between a set of design descriptions and a set of evaluation criteria. (We might think of this as applying evaluation functions to designs.) The inverse of this is a mapping from points in the space of evaluations to sets of designs, and this is how we see the activity of design: iteratively choosing a set of evaluation criteria and then finding sets of designs that are elements of this inverse mapping. We call these inverse mappings *design relations*. (They are relations, rather than functions, because they are in general one-to-many mappings.) The general sources for such relations are:

- The domain theory, i.e., in addition to evaluation functions some design relations may be known.
- Deduction from evaluation functions; some evaluation functions may have such a form that it is possible to obtain the inverse relation mathematically.
- Through search; in the absence of complete knowledge, it may be necessary to heuristically search through a range of possible designs in a generate-and-test paradigm. This search is guided by the design process model.

Three situations can occur:

- The inverse mapping does not exist, i.e. the design problem is unsolvable because it is overconstrained;
- There are many possible inverse maps, i.e. the design problem is underconstrained;
- There is a unique inverse, i.e the design problem has exactly one solution.

A basic design process, then, is to first find an estimated level of constraint, then to alternate between processes that

- adjust the degree of constraint (possibly using preferences) to moderate the ease or difficulty of finding solutions, and
- engage in search (or other methods of creating inverse mappings) to establish a set of acceptably-constrained design alternatives.

5 Unit Design

Unit design in ACTION is carried out as a highly-constrained search through a space of alternative unit designs. This is a heuristically guided exploratory search for several reasons. First, we have incomplete knowledge with which to design units algorithmically, and so unit design requires exploration, which must be to some extent a generate-and-test search process. Heuristics such as user priorities for goals or particular attributes, and user choice preferences are brought to bear during this search.

Second, the ability to explore and revise tentative conclusions opens the door to creative and possibly uniquely good, yet unforeseen, unit designs. This search through alternatives is thus important to maintain ACTION's capability to generate innovative and exploratory unit designs that nonetheless reflect solid, rational, and explainable design principles.

Search depends upon the ability efficiently to generate alternatives and the ability efficiently to test and differentiate promising avenues from unpromising ones. Thus search requires 1) good control (generation) heuristics, and 2) good evaluation heuristics---i.e., several good evaluation procedures graded in terms of resources required and solution quality, so that good and poor solutions can be differentiated rapidly, and so that greater effort can be put into detailed evaluation of highly promising solutions.

Overall unit design in ACTION requires designing two aspects: individual units themselves, and the set of relationships among units which, taken together, we call *unit structure*. There is no necessary commitment to which of these aspects is designed first. In ACTION Phase I, however, unit structure is constrained to be a hierarchy. This unit structure constraint means that at each hierarchical level, unit design can be seen as a problem of partitioning the collection of all activities at that level into a set of separate, disjoint units

5.1 Equivalence Relations as a Foundation for Unit Design

Unit design is based on the notion of equivalence classes, where units comprise activities that are found to be equivalent with respect to some equivalence criterium. The idea is that activities are put together in units because they are 'equivalent' in some sense, e.g. they may require the same resources or skills. By changing and composing the criterium for equivalence, ACTION can change the classification of activities into equivalence classes, and thus can manipulate the membership of activities in units. Since there is a number of different criteria, choices about *how* to construct an equivalence relation effectively yield a search through a space of alternative unit designs. The basis of this procedure is the fact that equivalence relations between objects in a set mathematically define a clustering into disjunctive clusters.

Using this approach, the issue becomes how to utilize several types of relations present in the ACTION domain theory to produce the "right" definition of equivalence and hence the right clustering.

The relations needed by this approach are found in the ACTION domain theory as factors such as:

- Subdivisibility of goals
- Reciprocal dependencies among tasks
- Motivational completeness of jobs
- Complete variance control
- Similarity of Information and Resource Needs
- Preference for job broadening
- Workload
- Needed vs. available skills
- Preference for single person jobs

These factors define relations between activities, but do not necessarily define equivalence. However, it turns out that even if a factor does not define an equivalence relation, it is usually possible to redefine the factor slightly, in a way that is inconsequential from the domain theory point of view, to make it an equivalence.

5.2 Unit Design Heuristics

There are two kinds of failure modes for a test evaluation, corresponding to the two dimensions of organizational evaluation used in ACTION; sociotechnical match and achievement of organizational goals. In the first case, a unit design may lead to an overconstrained set of organizational attributes (no sociotechnical match is possible among organizational features). In the second, the organizational feature matches may be underconstrained, but it may not be possible to fully meet organizational goals.

Unit design generation heuristics correspond to the failure modes of evaluations used in ACTION. For overconstrained attribute matches, these include ranking of equivalence criteria, (impacting what criteria are included or eliminated first, and thus which activities appear in which units) ranking of organizational goals (e.g., impacting core work scopes), and ranking of importance of organizational attributes (e.g., impacting what gets compromised first) for productive revision in overconstrained situations. For inability to meet organizational goals, these include ranking organizational attributes to manipulate criteria of acceptability.

6 Implementation Status

ACTION is implemented in Common Lisp, Garnet (a GUI tool), and X-Windows, and comprises approximately 2MB of application source code. As of this writing (May 1993) the ACTION software has gone through 6 major versions and over 40 minor versions in a development span of approximately 11 months, as part of its current design and development process. The software has been integrated and operational since August 1992, and has been in use and under pilot test on analyses of real manufacturing organizations since February 1993, and is being used in three manufacturing organizations.

7 Conclusions

The ACTION theory that forms the core of the knowledge in the ACTION decision support tool is based on an innovative open-systems approach to modeling the interactions among a wide array of TOP features in organizations. In addition, several innovative and theoretically strong approaches for knowledge representation, evaluation, design, explanation, and system generation. These include:

- Shared, replaceable knowledge representations for many system functionalities at several abstraction levels
- Constraint propagation for unit design and evaluation
- The use of equivalence relations as a foundation for unit design
- Direct representation of domain theory as a foundation for explanation
- System generation through incremental translation and aggregation of replaceable parts.

Overall, the ACTION project is an attempt to meld state-of-the-art knowledge representation, reasoning, control, and user-interaction structures into a usable and useful system for analysis and design of real human organizations. A second focus of ACTION is the development and synthesis of a robust and general theory of organization, with special attention to the integration of technological, organization-level, and human-level features. We expect that this theory will generalize to 1) other organizational contexts beyond production areas of discrete parts manufacturing (we are currently exploring applications in environmentally-conscious manufacturing and in software development organizations), and 2) non-human organizations, including distributed AI and multi-agent software systems. Finally, ACTION aims to provide a flexible and domain independent constraint-based modeling and reasoning structure that can be of significant value for development of many types on flexible theory-driven knowledge systems.

8 Acknowledgment

This paper was produced as a result of a program sponsored by the National Center for Manufacturing Sciences (NCMS).

9 References

[Erman 90] Lee D. Erman, ed. "Intelligent Real-Time Problem-Solving Workshop Report." Technical Report, TTR-ISE-90-101, Cimflex Teknowledge Corp., 1810 Embarcadero Road, P.O. Box 10119, Palo Alto, CA. 94303, 1990.

[Gasser and Majchrzak 92] Les Gasser and Ann Majchrzak, "HITOP-A: Coordination, Infrastructure and Enterprise Integration," in *Proceedings of the First International Conference on Enterprise Integration*, MIT Press, June, 1992.

[Majchrzak et al. 91] Ann Majchrzak, Mitchell Fleischer, Dave Roitman, and Joan Mokray, *Reference Manual for Performing the HITOP Analysis*. Industrial Technology Institute, Ann Arbor, MI, 1991.

[Majchrzak 88] Ann Majchrzak, *The Human Side of Factory Automation*, Jossey-Bass, San Francisco, 1988.

[Majchrzak and Gasser 92] Ann Majchrzak and Les Gasser, "HITOP-A: A Tool to Facilitate Interdisciplinary Manufacturing Systems Design," *International Journal of Human Factors in Manufacturing*, 2:3, 1992.