SSOAR

Open Access Repository
www.ssoar.info

Artificial intelligence modelling: data driven and theory driven approaches

Manhart, Klaus

Veröffentlichungsversion / Published Version
Sammelwerksbeitrag / collection article

Empfohlene Zitierung / Suggested Citation:
Manhart, K. (1996). Artificial intelligence modelling: data driven and theory driven approaches. In K. G. Troitzsch, U. Müller, G. N. Gilbert, & J. E. Doran (Eds.), *Social science microsimulation* (pp. 416-431). Berlin u.a.: Springer. https://nbn-resolving.org/urn:nbn:de:0168-ssoar-58475

Nutzungsbedingungen:
Dieser Text wird unter einer Deposit-Lizenz (Keine Weiterverbreitung - keine Bearbeitung) zur Verfügung gestellt. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.
Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use:
This document is made available under Deposit Licence (No Redistribution - no modifications). We grant a non-exclusive, non-transferable, individual and limited right to using this document. This document is solely intended for your personal, non-commercial use. All of the copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute or otherwise use the document in public.
By using this particular document, you accept the above-stated conditions of use.

gesis Leibniz-Institut für Sozialwissenschaften

Mitglied der Leibniz-Gemeinschaft

19 Artificial Intelligence Modelling: Data Driven and Theory Driven Approaches

Klaus Manhart
Universität Leipzig
Institut für Soziologie
Leipzig, Germany

19.1 Three Dilemmas of Conventional Computer Models

Building Theories is the central aim of empirical science and the computer is a valuable supporting tool for doing this. On the one hand, existing theories can be modelled precisely in form of computer programs, on the other hand theories and explanation facilities can develop from modelling empirical data by and by. Beyond traditional empirical methods, computer simulation is regarded more and more as the second pillar of empirical science and key technology for the next decades.

However, the powerful theoretical potential of computers contrasts to their factual use in the empirical sciences. Empirical scientists view computers mostly as number crunchers doing arithmetical operations on figures. Especially in the humanities and social sciences the dilemma arises, that most theories exist in verbal form and cannot be translated into numerical, conventional models as used by natural scientists (for example differential equations).

The first and most important problem with social science computer modelling therefore is, that social science knowledge often has qualitative, non-numerical form, which cannot be described by numerical algorithms of tradi-

tional programming techniques. For example, in the social sciences there are typically qualitative relations like:

- if x then y
- if x then not y
- the bigger x the bigger y.

Conventional modelling demands specifying these relations in exact, quantified terms. This means either to state social science knowledge "precisely", but in an artificial and brute force manner or thus not to be able to apply it to computer programs. This is the first and most important dilemma of social science knowledge in the frame of numerical modelling.

A second problem with conventional computer models is that theoretical knowledge often is hidden in non-theoretical code. In every model there has to be code elements for program control, for user interface, for reading data and so on. These parts have no relation to the theoretical knowledge. In conventional programs a strictly separation of domain specific knowledge and other program elements is to carry out very difficult. The reason is that in conventional programming methods a strict defined interface between domain specific and non domain specific parts is lacking. Various authors, e.g. Frijda [1967], have emphasized that this partition in computer programs is very difficult to obtain.

Boden [1984] refers to a third problem of conventional models: She writes, that the reasoning – or rather the calculation – behind the conclusions generated by these numerical models is commonly opaque to all but the mathematical initiate. This makes it very difficult for politicians and the general public to produce sensible critiques of the inferences and assumptions involved, or even to understand such critiques when produced by other social scientists.

19.2 AI based modelling

In the eighties a few sociologists, social psychologists and anthropologists have discoverd the power of Artificial Intelligence (AI) for avoiding or solving these problems (for example Brent 1986; Sylvan and Glassner 1985; Read and Behrens 1989; for an overview and general introduction: Benfer *et al.* 1991). AI offers concepts, techniques and tools to reduce the three dilemmas and leads to a lot of additional advantages. The general AI-modelling view is the view of computers as symbol processing machines doing operations on any symbol. The key term is the concept of knowledge representation. Knowledge

representation means the formal representation of qualitative, semantic, non-numerical information. For modelling uses, knowledge representation has three dimensions:

1. knowledge representation as abstract concept means formalisms like logic or higher order structures such as semantic nets, frames or scripts (for an overview see e.g. [Rich 1983]). In our context, especially first order logic is a very important representation formalism because nearly any qualitative theory can be translated into this formalism [Kowalski 1979].

2. declarative programming tools and techniques for implementing knowledge on a concrete machine like Lisp [Winston and Horn 1981], Prolog [Clocksin and Mellish 1984] or higher order tools, for example hybrid systems.

3. knowledge based systems or expert systems as a possible architecture with well–known features and advantages (e.g. Waterman 1986; Merritt 1989).

The focus of my interest is knowledge based systems. Knowledge based systems or expert systems are computer programs, which represent and process the knowledge of a human expert. In these expert systems knowledge is explicitly coded, mostly in form of facts and if-then-rules. Since rules are the dominant formalism for representing expert knowledge, these programs are also called rule based systems.

The architecture of a rule based system consists at least of three parts [Merritt 1989]:

1. a knowledge or data-base which contains the qualitative data of a certain knowledge domain;

2. an inference engine to deduce new knowledge from the old one;

3. an explanation part, which shows the user, which facts and rules was used for deduction. The explanation part makes the action and internal structure of the system transparent.

The relation between scientific theories and knowledge based systems is as follows: knowledge based systems represent the domain-specific knowledge of human experts. Qualitative theories also contain domain-specific knowledge and scientists, working with a theory, can be regarded as experts on this theory. Therefore, knowledge based programs are the preferable form of computer models to represent these qualitative theories. In this case, we talk of knowledge based models. Thus, a knowledge or rule based model is a computer program, which represents some aspects of a real system or a theory in symbolic, rule

form. The knowledge base contains the social science knowledge and the inference engine deduces new facts from the old one. A fully implemented expert system can even explain, how these facts are deduced, e.g. which rules are used for deduction. This type of symbolic programs expands social science modelling in a decisive manner and releases it from the straitjacket of traditional modelling concepts. The following points emphasize important solutions for the above mentioned dilemmas.

- Symbolic, non-numerical representation of knowledge
  In expert systems, there is no need for quantification and no need for the usage of numerical concepts. In view of the specific form of social science theories, this modelling concept offers a wide area of applications. The most important motivation for the use of knowledged based models was therefore the possibility to formalize and explicate verbal theories which are beyond quantitative mathematical modelling. For Brent [1986] the greatest appeal of these programs is their claim to be applicable to problems not having tractable solutions based on mathematical reasoning.

- Separating domain specific code from other program elements
  The architecture of a knowledge based system implies a clear separation of the domain-specific knowledge from other program elements. Regarding modelling, this partition corresponds to a division of theoretical knowledge from other parts of the programs serving as control, interface et cetera. Thus, it can be definitely determined which program elements represent theoretical parts.

- Modularity and flexibility
  The consequence of this strict separation is a high degree of modularity and flexibility of the theoretical part. This means, that pieces of knowledge or modules simply can be added to the existing theoretical knowledge by adding rules, without changing other parts of the progam. In the same manner, pieces of the model can be modified or deleted easily by modifying or deleting certain rules. This has a very positive influence on the experimental character of computer modelling and theory construction.

- Explanation facilities
  Knowledge based systems have the ability to explain their reasoning. Computer models, which are able to explain their behavior, are important for validation, because the rules and assumptions can be shown and criticized. The expert system reasoning can be used for proofing the correctness of deductions or for showing the solution path. In general, better insights into the functioning of the model are possible. This reduces the accusation that computer models are opaque.

- Explication of logical features
  Automated inferencing in knowledge based models can eliminate the pitfalls of intuitive reasoning and flaws of verbal theory building. These flaws stem primarily from the limits on the processing capacity of the human brain like the small short term memory or various biases that may distort the inference process. For example, theories may hinge on the researcher's values, information is processed selectively, "desirable" conclusions are more easily drawn. Contrary to the intuitive, brain-based inferencing, computer-based inferencing can be made reliable and reproducible. In contrast to human scientists, knowledge based systems can deduce complete and correct conclusions from a big mass of data. This means concretely: The set of *all valid propositions* can be derived from given premises. The consistency of a theory can be proven by deriving no contradictions. The *soundness* of postulated conclusions can be shown. *New hypotheses* can be derived.

For Brent [1986], the expert system may play a role analogously to that of a good research assistant, checking the logic of the researcher, tracing out implications of changing an assumption, answering questions about its reasoning, and identifying the source of particular bits of knowledge. Of course, an AI approach has not only advantages but has also some difficulties and problems. For example, converting verbal model descriptions into symbolic terms is far from easy in general. In practice, the implementation of professional expert systems is often supported by a special "knowledge engineer" who transforms the verbal and vague expert knowledge to formal if-then-rules. Further, in complex problems it can be difficult to follow the reasoning of an expert system, so the user can be confused in the same way as in traditional models.

19.3 Theory driven AI-modelling

There are two fundamental ways to use AI-modelling for theory building. So far, a view of computer modelling was implicitly presented, which Herbert Simon calls theory driven science. Theory driven science means, that the starting position is an existing theory which should be represented in some or all aspects in form of a symbolic model. In one word: one tries to transform a theory to a runable program. If one is successful, the program represents the theory or parts of it more or less. In a pure theory driven approach, empirical data are not significant.

Why should we make computer programs out of theories? We should do this, because there are some decisive advantages to have a theory modelled

in form of a program, some mentioned above. To give only a few further key words: formalization, executability, experimental power, teaching. One of the most important reasons for modelling a theory is scientific discovery: in the theory driven approach, this means for example, the discovery of new theoretical conclusions, gaps, or inconsistencies. However, while theory driven science in traditional modelling was restricted to a high degree to numerical concepts, the knowledge based approach frees social science modelling from these restrictions. The knowledge based approach implies a valuable, more flexible and more applicable instrument of theory driven modelling. Let's illustrate the theory driven approach with a few concrete examples.

A very early, but prototypical example is Gullahorn and Gullahorn's [1963] computer model of Homans' interaction theory. Gullahorn and Gullahorn translated Homans not in expert system form, but into a qualitative computer program using IPL-V, an early Lisp-like language. One could say, that the program represents Homans theory to a big part.

In the eighties, a lot of modellers rely explicitly on the knowledge based paradigm. Each of the following programs exemplifies important advantages and aims of the theory driven approach.

Sylvan and Glassner [1985] translated theoretical assertions of Simmel's "Conflict and the web of group affiliations" into a rule-based program consisting of 223 rules. They used the program to determine whether the theory is internally contradictory and whether the central tennet of Simmel's work – the conflict-cohesion hypothesis – can be deduced. It was shown, that the verbal assumptions of Simmel were not sufficient to deduce the hypothesis.

Brent [1986] describes a Prolog program which uses Goffman's dramaturgical framework to analyze various examples of social interaction. Goffman's model represents a kind of sociological knowledge which has resisted attempts at formalization with mathematical models. These programs, so Brent, may one day be used as teaching tools to acquaint students with particular theoretical perspectives, and as aids to assist in refining and extending social theories.

The research group of Michael Masuch (Center for Computer Science in Organization and Management, Amsterdam) translated various theories from organizational sociology to first order logic and if-then-rules. Using a theorem prover for deductions it was feasible to draw all possible inferences from the theory's assumptions. Glory *et al.* [1990] clarified the structure of Mintzberg's contingency theory and encountered some terminological problems at various points during the formalization process. Another example by Péli *et al.* [1994] is the logic translation of organizational ecology by Michael Hannan and John Freeman. The application of a theorem prover demonstrated, that the chosen part was consistent – with one provision. Further, a set of new theorems were discovered and some postulated conclusions could not be deduced from the assumptions.

It should be clear that these four examples represent a pure theory driven approach. All models start with explicitly existing theories and translate theoretical assumptions to qualitative program statements, partly discovering new and unknown theoretical and logical features. In all of these studies empirical data were not relevant.

19.4 Data driven AI-modelling

The counterpart of a top-down perspective is a bottom-up perspective: the data driven approach. In contrast to the theory driven procedure data driven science starts vice versa with empirical data or the input-output behavior of the real system without an explicitly given theory. The modeller tries to write a computer program which generates the underlying empirical data or input-output(i-o)-behavior of the system. Typically, models are produced in a generate-and-test-procedure. Generate-and-test means writing program code which tries to model the i-o-behavior of the real system first of all roughly. Using the generated code one tests the i-o-behavior and improves it as long as the i-o-behavior doesn't correspond to the real system.

The data driven approach is very popular in cognitive modelling, for example in modelling problem solving processes. Langley, Simon and others [1987, p. 33] give a short abstract of this procedure: "We can write in an information processing language – a symbolic rather than numerical language – a computer program that describes, say, the processes that intelligent adults are hypothesized to use in solving some class of problems. We can then present the program and some human subjects with identical problems and compare their behaviors. From the computer we will obtain a trace, from the human subjects a verbal and written protocol of their behavior while they were solving the problems. We can then test the program – our theory of the behavior – by comparing the trace with the protocol, just as any theory is tested by comparing the system path it predicts with data showing the actual path".

As the quotation shows, the goal of this approach is the discovery of mechanisms, of hypotheses or a theory, which describes the underlying phenomena. Permanent model improving and fitting will lead to more adequate models which fit the empirical system better and better. It will give insights into the complex empirical system and thus theoretical structures can develop. In my opinion, the data driven modelling is at least as useful as the theory driven approach. The reason is, that our existing, empirical confirmed theories can not be better than the best theories which come to our mind. Efforts to improve the conditions of theory building in the social sciences will therefore lead to better empirical

confirmed theories [Glory *et al.* 1990, p. 81]. Knowledge based data driven models will improve these conditions in a decisive and fruitful manner.

The Teamwork project by Doran [1985] can serve as the first simulation example for this approach. Teamwork is intended to permit the study of task planning and task execution and of the mechanisms of cooperation and communication between actors. Its aim is to use the study of this modelled multi-actor system to further understandings of real systems – "both those that might be constructed and those that are in existence around us". As far as can be seen, Doran doesn't use an explicitly given theory but he tries to model empirical phenomena by generating computer behavior, which corresponds to empirical data.

Banerjee [1986] uses a Prolog program to explain the development of social structures. For explaining social structures he uses the scheme concept from AI and cognitive psychology. He argues that social structures endure only when the preferences and judgments of participants prompt them to act in ways that reinforce the preferences and judgments of other participants. When actions of each set of participants reinforced the preferences and judgments of others, that particular social structure tended to reproduce itself and to be stable over time. This program can be regarded as an example where theoretical structures are developed from data and – as Benfer *et al.* [1991, p. 26] say – as "use of artificial intelligence programming strategies where the AI program becomes the theory itself".

The expression "development of theoretical structures from data driven computer programs" has a precise equivalent in Artificial Intelligence. Under the label "machine learning", computers are programmed to discover empirical laws from data [Langley *et al.* 1987; Slezak 1989]. For example, given the rough data by Robert Boyle, one program is reported to have no difficulty finding the law relating gas pressure and volume. A well-known family of these discovery models are the Bacon programs. Their results are published in papers like "Rediscovery Chemistry with the Bacon System" or "Rediscovering Physics with Bacon 3" [Langley 1979]. Bacon 3 is a knowledge based system using a few simple heuristics to solve a broad range of tasks. The rules detect constancies and trends in data and lead to the formulation of hypotheses. Bacon 3 has shown its generality by rediscovering versions of the ideal gas law, Kepler's third law, Coulomb's law, Ohm's law and more. The discovering procedures are not limited to quantitative laws. For example, Glauber discovers qualitative regularities in chemistry.

By the way, these programs demonstrate that apparently there exists a logic of inductive reasoning - in contrast to what Popper says. In any case, for the social sciences the careful hypothesis can be stated: inductive learning programs could be implemented for the discovery of regularities in social science data too. However, these systems must be able to handle statistical laws. A first beginning

is maybe Garson's paper "The role of inductive expert system generators in the social science research process" [Garson 1987]. Garson successfully applies a classification and rule generating algorithm to historical data matrices.

Until now, the concepts of theory driven and data driven science were introduced as ideal cases. However, in practice the dividing line between theory driven and data driven approaches is often not as clear as it seems here but more fuzzy or mixed. For example, one can build-in (parts of) a theory in a data driven procedure or vice versa or one can go back and forth from theory to data and so on.

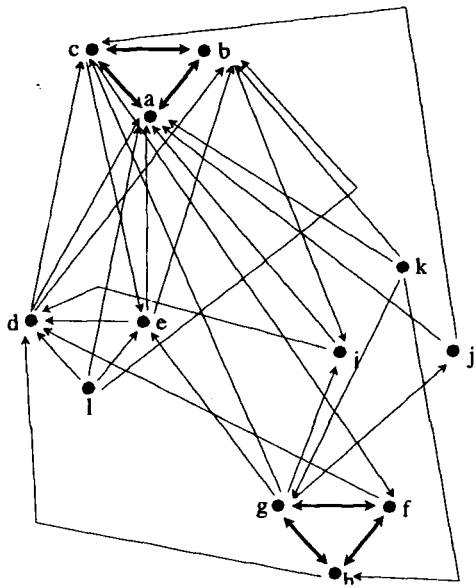
19.5 A mixed approach: discovery of theoretical explanations for group processes

In this last section, an example of the author's work is given. It's an instance for a mixed approach, where the procedure is the following: starting with a model of a rather vague theory, the consequences of the model inputs were investigated. The computational data, which were produced by this model, were compared to empirical data. To make a better fit, the model was changed, run again and started up again. In this back and forth procedure the goal was to discover new und specific hypotheses and explanations for empirical phenomena which the theory dealt with.

The modelling object was a micro-sociological structural theory: the transitivity model by Holland and Leinhardt [1971]. Generalizing Heider's balance theory, the transitivity model deals with interpersonal relations in small groups, especially friendship choices. The central claim is that relations in these groups become more transitive over time and – as a consequence – lead to a hierarchical structuring of cliques. Transitivity tendency means, that if x is a friend of y and y is a friend of z then – on the long run – x will also become a friend of z. Fig. 19.1 is an example of a potential friendship network, represented by a graph structure. The nodes represent actors, and the arrows (or arcs) "directed relations" between the actors. Interpreted as a friendship network, the arc pointing between a and b means, that a is a friend of b. Fig. 19.1 is an intransitive structure, because transitivity does not hold for all members of this network. For example, k is a friend of h and h a friend of f, but k is not a friend of f as the transitivity rule requires. Note, the elements of a sub-graph, in which all members choose all other members, are called cliques. Thus, $\{a, b, c\}$ and $\{f, g, h\}$ are two cliques.

The transitive graph model was confirmed by a lot of studies (for example Davis 1970; Hallinan 1974). In the original model however, there is a gap of how

Figure 19.1. An example of an intransitive network structure



these graphs become more transitive and why. Although the theory is supported by empirical data, it gives no explanation for the development of transitive structures. Further, the empirical content of the theory is very low. Hummell and Sodeur [1987] for example have criticized that there are no mechanisms that could explain the emergence, development, stabilization and changing of transitive structures. Opp [1984] criticizes the low empirical content of the theory, which means, that the development of more transitive structures is truly predicted, but these structures can be realized in a lot of alternatives. There is no prediction, how the transitivity tendency will be developing concretely. However, the more the number of alternatives, the lower the content of a theory.

In a computational approach, an attempt was made, to detect and investigate interesting theoretical hypotheses, which should explain the development of transitivity in friendship choices and should increase the empirical content. This was done by changing from theory to data and vice versa. These kind of computer experiments can precede and support empirical research because they can clarify the hypotheses and discover formal and substantial interesting features of a theory. For example, some assumptions are maybe not very sensible, while others can be interesting candidates for empirical investigations. Thus, computer experiments can be an important kind of formalized and controlled thought experiments, which support empirical research.

A Prolog program was written that firstly represents the essential concepts

of the theory in the architecture of an expert system. It has the ability to explain its reasoning and to trace the chains of reasoning. Secondly, step by step procedures were implemented, which transformed intransitive networks into transitive structures. Logically, these procedures are equivalent to a specialization of the qualitative Holland and Leinhardt law of transitivity tendency. The structural consequences of these procedures were observed and compared to empirical data. If the produced consequences did not fit the qualitative data, they were varied to make a better fit.

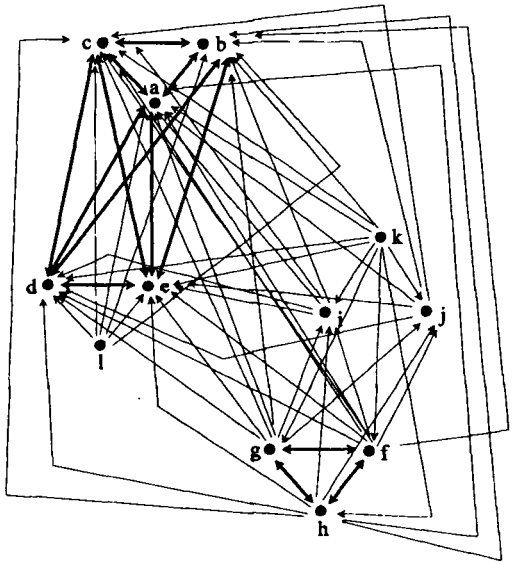
Formally, there are two fundamental ways to increase transitivity in an intransitive structure. Assume, $r(x, y)$ and $r(y, z)$ are two existing relations in a (friendship) database but $r(x, z)$ is missing – in contrast to the requirement of the transitivity rule. A first approach for generating more transitivity is simply to add the relation $r(x, z)$ to the knowledge base (Add-Rule). Thus, transitivity could be generated by checking all triples $\langle x, y, z\rangle$ and adding $r(x, z)$ whenever $r(x, y)$ and $r(y, z)$ does hold. A second approach is to delete one of the relations $r(x, y)$ or $r(y, z)$ in the data base. In this case, the if-part of the transitivity rule does not apply and the rule is not violated. Thus, "vacuous" transitivity could be generated by checking all triples $\langle x, y, z\rangle$ and delete $r(x, y)$ or $r(y, z)$, whenever these two relations are in the data base, but $r(x, z)$ is not included (Delete-Rule). Note: in both cases the changing (adding or deleting) of a relation has a potential effect on the transitivity of other triples.

These two basic procedures are very primitive, but can serve as potential starting position. First, the Add-Rule seems to be the one which es empirically preferred the most. In real group processes, transitivity is "resolved through transitive closure rather than through the development of vacuous transitivity" [Holland and Leinhardt 1971, p. 123]. Second, we assume cliques as a starting position for the correction process, because in these sub-groups the cohesion is extremely strong – compared to actors which are not clique members. Thus, the "transitivity pressure" is stronger in cliques than in non-clique-members.

By pressing the "run-button" with this specification of the Add-Rule, the simulation generated transitive structures. However, these structures did not fit the empirical data. Using different data sets in many runs, cliques were unified when making intransitive networks more transitive. For example, the separated cliques $\{a, b, c\}$ and $\{d, e, f\}$ in an intransitive starting network, can be unified to the clique $\{a, b, c, d, e, f\}$ in the generated transitive structure. A closer look shows that cliques are combined whenever there are cycles between cliques. A cycle between clique A and clique B is a directed path from a clique member of A to a clique member of B and back again to A. The pooling of cliques, however, is an unacceptable fact because empirically this doesn't occur.

The data implied that the transitivity mechanism should be refined. Thus, the constraint was added so not to pool existing cliques. This implied, not only adding relations, but removing relations, as the union of the cliques was

Figure 19.2. The model generated transitive structure of the intransitive graph in Fig. 19.1



threatened. Further, the data suggested, that the transitivity machine should start with cliques of high status order. A clique $C1$ has a higher status than a clique $C2$ if and only if more actors voted for the members of $C1$ than for the members of $C2$ – or graph theoretical: if more arcs point at $C1$ than at $C2$. In this manner, the generation of a set of theoretical interesting procedures was continued which had as a consequence transitive structures. These procedures fit the empirical data better and better. Fig. 19.2 shows a generated transitive structure using the data of Fig. 19.1. As a result, a hierarchical transitive structure was produced with two cliques $\{a, b, c, d, e\}$ and $\{g, h, f\}$. Clique $\{g, h, f\}$ has a very low status, whereas clique $\{a, b, c, d, e\}$ has a very high status. Note: for every member of the network the transitivity rule holds. The so-produced transitivity mechanism has powerful explanation facilities for important group processes. They can serve as structural explanations for dynamic clique development. In detail, the computer based transitivity model generates and explains the following tendencies:

(a) Destroying of existing cliques

Although cliques are very cohesive sub-structures and will usually not be destroyed in the specialized model too, in some cases they are disbanded. On the way to more transitivity, cliques will be destroyed if and only if two or more cliques are connected via co-cliqual nodes. A co-cliqual node is an actor, who is a member of at least two cliques. Which clique

will be destroyed depends on the status of the clique. For example, in the four 2-cliques in Fig. 19.3 two of them are disbanded and only the cliques with the highest status stay alive.

(b) Integrating new members into existing cliques
On the way to more transitivity, cliques will expand their number of members, if the non-members are connected to these cliques by cycles. For example, in the intransitive graph of Fig. 19.1 the non-clique-actors d and e are joined with the clique $\{a, b, c\}$ by a cyclus (e.g. $e - a - c - e$ or $d - c - e - d$). By making this graph more transitive, there is a tendency of integrating d and e to the clique, and this dynamic process is finished in a complete transitive graph.

(c) Emergence of new cliques
On the way to more transitivity, cliques will emerge from those actors, who are connected via cycles. Every cycle in an intransitive graph will be completed to a clique in a transitive graph. Thus, the cycle on the left side of Fig. 19.4 will become a clique in a transitive structure. Metaphorically speaking, the cycle is the "egg cell" for the emergence of cliques when using the structure model of the transitive graph.

Figure 19.3. Chains of cliques will be destroyed when the graph becomes transitive.

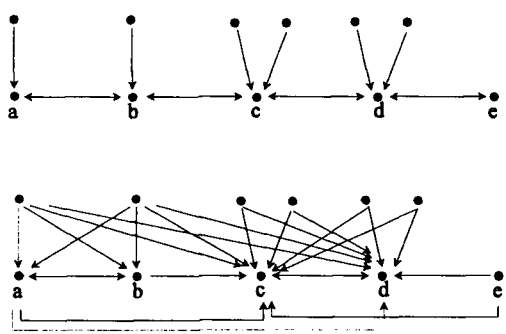


Figure 19.4. A cycle in an intransitve structure (left side) will be completed to a clique in a transitive structure (right side).

The experiments with the computer program indicate a very important role for cycles in the transitivity theory and in the explanation of group processes.

A cycle is a critical graph theoretical concept in this model and a sensitive theoretical feature of the Holland-Leinhardt theory.

To summarize, the following steps were carried out:

1. Some logical possibilities, how graphs become more transitive, were implemented in rule-form;

2. Regarding empirical data, these transitivity rules were modified and made "empirically more adequate";

3. The result was a transitivity rule which could explain a number of dynamic group processes such as clique developing, clique expanding and clique destroying.

The discovered mechanisms are not inductively reasoned directly by the program, but by experimenting with the program. They can be viewed as research hypotheses, which are candidates for empirical investigation.[1]

19.6 Conclusion

In general, AI-techniques can both be used for investigating and improving existing theories and for generating new and better theories from empirical data. AI can support the process of scientific development analogously like some formal aids in mathematical modelling are doing. A lot of authors emphasize AI-modelling as being much more significant in cultural and social studies than, for example, modelling by mathematical equations. An extreme position states that AI will take the same role in the social sciences which mathematics had in the physical sciences: "Artificial Intelligence will ultimately come to play the role vis-a-vis the psychological and social sciences that mathematics, from the seventeenth century on, has done for the physical sciences" [Allport 1980, p. 31].

Although we have argued for AI in this paper, we do not share a too uncritical enthusiasm. In fact, many researchers also deny a significant role that AI plays in the social sciences. Additionally, there are a lot of problems for applying AI and expert system techniques to social theories. Thus, a good strategy is a moderate optimism. In our view, AI is not a revolution – as Allport's proposition suggests – but rather an evolution in the development of new modelling techniques. Beyond some very useful parts of mathematics,

[1] By the way, eliminating the knowledge base – i.e. facts and rules of the Holland-Leinhardt-theory – the program can be used as an expert system shell for implementing other theories. For further details see Manhart [1995].

statistics and traditional computer modelling, AI is one new tool to formalize and state more precisely social science knowledge. We will do best, if we select from the modelling tool box those instruments which are most adequate for our purposes. Presumably, AI will at least play a significant and essential role in the competition of modelling tools in the future.

Acknowledgment

This is a slightly revised version of parts of my PhD thesis. I thank two anonymous reviewers for their helpful comments and advices.

References

Allport, D.A. (1980) Patterns and Actions. In: G. Claxton ed. *Cognitive Psychology: New Directions.* Routledge & Degan, London, pp. 26–64.

Banerjee, S. (1986) Reproduction of Social Structure. An Artificial Intelligence Model. In: *Journal of Conflict Resolution.* 30, 2, pp. 221–252.

Benfer, R.A., Brent, E.E., and Furbee, L. (1991) *Expert Systems.* Sage, Newbury Park CA.

Boden, M. (1984) Artificial Intelligence and Social Forecasting. In: *Journal of Mathematical Sociology.* 9, pp. 341–356.

Brent, E.E. (1986) Knowledge-Based Systems: A Qualitative Formalism. In: *Qualitative Sociology.* 9, 3, pp. 256–282.

Clocksin, W.F., and Mellish, C.S. (1984) *Programming in Prolog (2.ed.).* Springer, Berlin.

Davis, J.A. (1970) Clustering and Hierarchy in Interpersonal Relations: Testing two Graph Theoretical Models on 742 Sociogramms. In: *American Sociological Review.* 35, pp. 27–33.

Doran, J. (1985) The Computational Approach to Knowledge, Communication and Structure in Multi-Actor-Systems. In: Gilbert, G.N., and Heath, C. eds. *Social Action and Artificial Intelligence.* Gower, Aldershot, pp. 160–171.

Frijda, N.H. (1967) Problems of Computer Simulation. In: Dutton, J.M., and Starbuck, W.H. eds. *Computer Simulation of Human Behavior.* Wiley, New York, pp. 610–618.

Garson, G.D. (1987) The Role of Inductive Expert Systems Generators in the Social Science Research Process. *Social Science Microcomputer Review.* 5, 1, pp. 11–21.

Glory, J.C., Masuch, M., and Marx, M. (1990) Formalizing Organizational Theory: A Knowledge-Based Approach. In: Masuch, M. ed. *Organization, Management, and Expert Systems. Models of Automated Reasoning.* de Gruyter, Berlin, pp. 79–104.

Gullahorn, J.T., and Gullahorn, J.E. (1963) A Computer Model of Elementary Social Behavior. In: *Behavioral Science* 8, pp. 354–362.

Hallinan, M. (1974) A Structural Model of Sentiment Relations. In: *American Journal of Sociology.* 80, pp. 364–378.

Holland, P.W., and Leinhardt, S. (1971) Transitivity in Structural Models of Small Groups. In: *Comparative Group Studies.* 2, pp. 107–124.

Hummell, H.J., and Sodeur, W. (1987) Triaden- und Triplettzensus als Mittel der Strukturbeschreibung. In: J. van Koolwijk and Wieken-Mayser, M. eds. *Techniken der empirischen Sozialforschung Bd. 1. Methoden der Netzwerkanalyse.* Oldenbourg, München, pp. 129–161.

Kowalski, R. (1979) *Logic for Problem Solving.* North Holland, New York.

Langley, P. (1979) Rediscovering Physics with BACON.3. In: *Proceedings of the Sixth International Joint Conference on Artificial Intelligence.* Vol.1, Tokyo, pp. 505–507.

Langley, P., Simon, H.A., Bradshaw, G.L., and Zytkow, J.M. (1987) *Scientific Discovery: Computational Explorations of the Creative Processes.* MIT Press, Cambridge, Mass.

Manhart, K. (1995) *KI-Modelle in den Sozialwissenschaften. Logische Struktur und wissensbasierte Systeme von Balancetheorien.* Oldenbourg, München.

Merritt, D. (1989) *Building Expert Systems in Prolog.* Springer, Berlin.

Opp, K.D. (1984) Balance Theory: Progress and Stagnation of a Social Psychological Theory. In: *Philosophy of the Sociological Sciences.* 14, pp. 27–50.

Péli, G., Bruggeman, J., Masuch, M., and Nuallain, B. (1994) A Logical Approach to Organizational Ecology: Formalizing the Inertia-Fragment in First-Order Logic. In *American Sociological Review* 59, pp. 571–593.

Read, D., and Behrens, C. (1989) Modeling Folk Knowledge as Expert Systems. In: *Anthropological Quarterly.* 62, pp. 107–120.

Rich, E. (1983) *Artificial Intelligence.* McGraw-Hill, New York.

Slezak, P. (1989) Scientific Discovery by Computer as Empirical Refutation of the Strong Programme. In: *Social Studies of Science.* 19, pp. 563–600.

Sylvan, D., and Glassner, B. (1985) *A Rationalist Methodology for the Social Sciences.* Blackwell, New York.

Waterman, D.A. (1986) *A Guide to Expert Systems.* Addison-Wesley, Reading, Mass.

Winston, P.H., and Horn, B.K.P. (1981) *Lisp.* Addison-Wesley, Reading, Mass.