

# I-X and <I-N-CA>: an Architecture and Related Ontology for Mixed-initiative Synthesis Tasks

Austin Tate

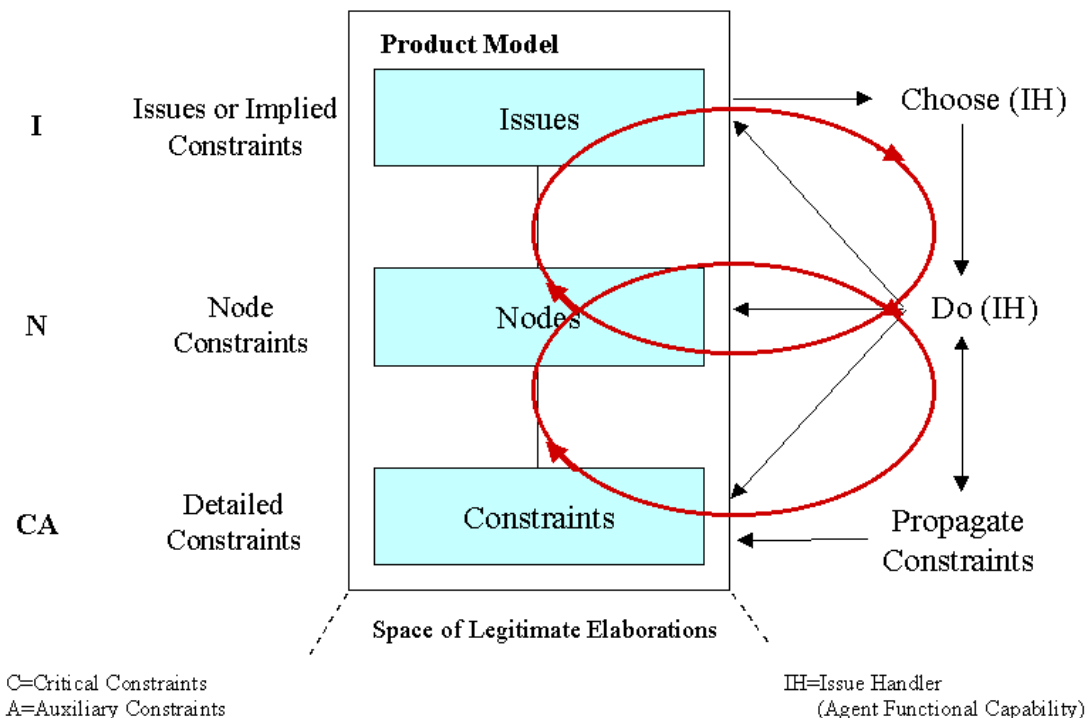
Artificial Intelligence Applications Institute  
 Centre for Intelligent Systems and their Applications  
 Division of Informatics, The University of Edinburgh  
 80 South Bridge, Edinburgh EH1 1HN, UK  
 a.tate@ed.ac.uk

**Abstract.** I-X is a research programme with a number of different aspects intended to create a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product such as a plan, design or physical entity – i.e. it supports synthesis tasks.

The I-X approach involves the use of shared models for task directed cooperation between human and computer agents who are jointly exploring (via some processes) a range of alternative options for the synthesis of an artifact such as a design or a plan (termed a product).

The <I-N-CA> (Issues – Nodes – Critical and Auxiliary Constraints) ontology is used to represent a product as a set of constraints on the space of all possible products in the application domain. The modular I-X systems integration architecture encourages the creation of systems as components which handle “Issues” related to the design and its requirements, selects “Nodes” as the principal entities to be incorporated into the design, and checks or maintains a set of constraints of various types.

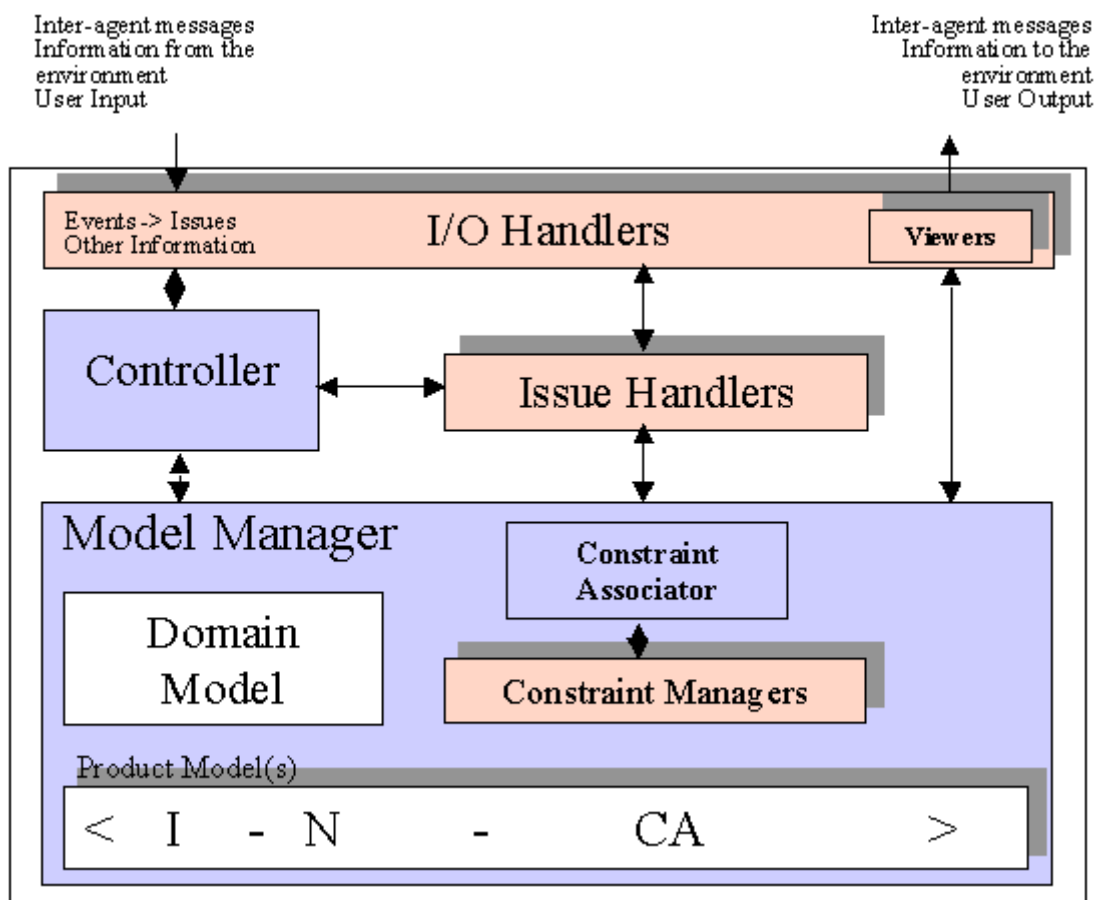
## 1 I-X Approach



The I-X approach involves the use of shared models for task directed cooperation between human and computer agents who are jointly exploring (via some processes) a range of alternative options for the synthesis of an artifact such as a design or a plan (termed a product).

- An I-X system or agent has two cycles:
  - Handle Issues
  - Respect Domain Constraints
- An I-X system or agent carries out a (perhaps dynamically determined) process that leads to the production of (one or more alternative options for) a synthesised artifact.
- An I-X system or agent views the synthesised artifact as being represented by a set of constraints on the space of all possible artifacts in the domain.

I-X also involves a modular systems integration architecture that strongly parallels and supports the abstract view described above. More detail is available on-line at <http://www.aiai.ed.ac.uk/project/ix/>.



## 2 I-X Research Programme

I-X is a research programme with a number of different aspects intended to create a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product such as a plan, design or physical entity – i.e. it supports **synthesis tasks**.

The I-X research draws on earlier work on O-Plan (Tate et. al., 1998; Tate et. al., 2000a; 200b). <I-N-OVA> (Tate, 1996; 2000a) and the Enterprise Project (Fraser and Tate, 1995; Uschold, et. al., 1998) but seeks to make the

framework generic and to clarify terminology simplify the approach taken, and increase re-usability and applicability of the core ideas.

Previous research was heavily applications driven or considered only support to certain types of task – especially Planning. The terminology and modules in the earlier systems therefore tended to reflect the applications being created and the types of task supported and were not sufficiently generic when a new application was approached.

The I-X research program includes the following threads or work areas, many of which were refined and further developed from the initial O-Plan and Enterprise approaches during the research supported by DERA:

1. **I-Core**, which is the core architecture, the underlying ontology of activity and processes termed <I-N-CA>, and the terminology used to describe applications, systems or agents built in the I-X framework.
2. **I-PE/I-DE**, which is the I-X Process or Domain Editor, which is itself an I-X application but also is used to create and maintain the process models and activity specifications used elsewhere.
3. **I-P<sup>2</sup>**, which are I-X Process Panels used to support user tasks and cooperation.
4. **I-Plan**, which is the I-X Planning System. This is also used within I-P<sup>2</sup> and other applications as it provides generic facilities for supporting planning, process refinement, dynamic response to changing needs, etc.
5. **I-Views**, which are viewers for processes and products, and which are employed in other applications of I-X. I-Views can be for a wide range of modalities and types of user.
6. **I-Faces**, which are underlying support utilities to allow for the creation of user interfaces (User I-Faces), repository access (Repository I-Faces), and communication with other systems and agents (Communication I-Faces).
7. **I-X Applications** of the above threads in a variety of areas depending on our current collaborations. These currently include:
  - a. Coalition Operations (I-DEEL)
  - b. Emergency, Unusual Procedure and Help Desk Assistance (I-Help and I-Rescue)
  - c. Multi-Perspective Knowledge Modelling and Management (I-AKT)
  - d. Natural Language Presentations of Procedures and Plans (I-Tell)
  - e. Collaborative meeting and task support (I-Room)

### 3 Overview of I-X and its Contribution

During initial studies, the I-X component definitions, purposes and interfaces were significantly clarified and made more general than previously. Some components previously considered as a key part of the design were called into question, and it was realised that they provided just one implementation style (albeit a useful template) for any of a wide class of approaches. Two such examples are:

1. The previous notion of user interface viewers has been simplified and can now admit much simpler interfaces more suited to embedded systems, as well as still supporting a style of user interface with explicit registered plug-in viewers for processes and products.
2. The previous belief that constraint managers would be plugged into the model manager via a “Constraint Associator” which would know the constraint types each could handle and needed to receive. This is now one style of implementation of an I-X model manager and not prescribed.

The design has thus become simpler in many respects, and allows for very simple implementations – as was always the intention. We have claimed that I-X can be used to mediate manual systems without any code involved at all (as in some Enterprise project applications that have had enormous commercial impact – e.g. at Unilever) and this is still the case. This can apply to the extreme extent that an I-X system can become a trivial to do list system and be manually enacted with almost redundant implementations of the controller (user picks next to do item), model manager (list of to do items and their status - done, not done) and Viewer (to do item string against a check box). However, demonstration applications created as part of the initial work have a lot more structure and underlying process knowledge and are driven by very general process models.

I-X can be considered to make contributions at 3 levels. These are:

1. An outer level approach of handling issues and respecting constraints in the domain model gives an intelligible approach to what an I-X system or agent does.
2. a middle level provides a fractally composable cell which employs a model/viewer/controller systems integration approach.
3. a detailed level provides an approach which represents and reasons with constraints on the space of all possible artifacts, and allows for the provision of specialised solvers for some or all of these detailed constraints.

I-X makes novel contributions at level 1 and 3, and is compatible with other approaches at level 2.

#### **4 History and Comparison of I-X to other Approaches**

I-X is based on ideas that have emerged in research on O-Plan since 1983 and the Enterprise project in the mid 1990s. At the time of its design in 1983, O-Plan (the Open Planning Architecture) was intended to offer a more flexible and modular way to build planning systems over those being created in the mid 1970s (Nonlin, NOAH, Deviser). In particular it sought to make the plan in a partially developed state be an entity that could exist separate to the planner. Previous planners maintained lists of flaws, interactions or “criticisms” that were to be fixed in their internal control data structures. This was made explicit and declarative in O-Plan using an “agenda” associated with each plan being developed.

Over the years the modules and components in O-Plan have been clarified and made more generic, a process accelerated by trying to use the design in other applications:

- for a scheduler TOSCA (The Open Scheduling Architecture) in a system for practical uses in Hitachi;
- for a system for planning and dealing with execution failures by repairing plans OPTIMUM-AIV for the European Space Agency;
- and especially for the Enterprise Architecture which took O-Plan ideas into business process management and support – including some manual process support at Unilever that has had enormous commercial benefits.

This re-use of the concepts led us to feel the original module or component definitions had their limitations, and their terminology was too orientated towards support to the planning task.

At the time of its design, O-Plan had similarities to the Blackboard Architectures (see for example Hayes-Roth and Hayes-Roth, 1979). O-Plan was specifically designed to avoid indirect aspects of that architecture in the way it mapped Knowledge Source Activation Records (KSAR) to the Knowledge Sources (KS) available in a Blackboard system. O-Plan had an agenda whose entries mapped quite directly to its Knowledge Sources. I-X terminology now refers to these as the List of Issues and Issue Handlers (so not committing to an agenda style of implementation). O-Plan also differentiated between general and maximally capably Knowledge Sources which could write any type of information into the Blackboard (where the plan or product was being created) and more limited special functionality embedded in “Constraint Managers” which could just check the information in the Blackboard/product and give yes/no/maybe answers on the validity of the Blackboard or product description. This information could then be used by other later Knowledge Sources to carry on processing. Decision-making in the system was thus located in Knowledge Sources and not the special limited Constraint Managers. As in Blackboard systems, the O-Plan controller dealt with ordering decision on which agenda entry to handle next – localising such control information too. These features enabled O-Plan to scale better to larger realistic tasks. This work was presented and the O-Plan approach contrasted to Blackboard systems at a scientific workshop on Blackboard architectures run by SPL Insight programme in the UK in the mid 1980s.

O-Plan also attempted to create a software engineering structure and systems integration framework for building practical planning systems. The design proved to have much in common with the contemporary work in the early 1980s at NASA and the National Bureau of Standards (NBS) on layered reference architectures for space station telerobotics and for flexible manufacturing cells. See for example the NASREM work (Albus et. al. 1987) and its later military version generated by Hayes-Roth (DICAM). This work has continued over the years with many

variants of systems that can take architecture “cells” and compose them in various ways, recursively, peer-to-peer and fractally. The most recent work on this was done in a working group on architectures by people engaged on many of these earlier architectures (Hayes-Roth, Tate, etc) in a group set up by DARPA to look at next generation Intelligent Battle Force Management which reported to DARPA in 1999.

O-Plan had a very optimistic and forward-looking target for its delivery platforms when first designed in 1983. It was designed for significant distributed systems implementations utilising:

- geographically distributed systems (perhaps far distributed with some elements in deep space and others being ground based with 8 hours message transmission times);
- symmetric local processors on which to mount parallel versions of the principal components of an O-Plan agent or system – allowing for multiple platforms to be running concurrently for dealing with agenda entries on one or more concurrently developed shared product model;
- fine grain parallel systems for constraint management, graph algorithm process and similar lower level functions. Implementations were even prototyped and fabricated in silicon to add in as specialised co-processors for constraint handling.

Much of the resulting structure and overhead for distributed implementations turned out to be far ahead of the time at which realistic implementation was possible, and this became a burden within the implementation. I-X allows for such sophistication in implementation, but does not clutter the elegance of the approach by embedding it into the interfaces and component boundaries designed.

The constraint representation of activity, <I-N-OVA>, did not emerge until the mid 1990s, although the components of its design had been in O-Plan since its inception. A terminology change to call agenda entries Issues was suggested by a DARPA programme manager (Craig Wier) in the early 1990s, and led to very profitable interaction with the engineering issue-based design community as a result. A joint interest in emerging standards for process modelling in IDEF-0 (and later IDEF-3) between DARPA, AIAI and ISX Corporation also led to clarifications of how a plan was represented in O-Plan and how it could be a basis for rich and enrichable process and activity representation standards. Discussions between the theoretical and practical AI planning communities (in particular Subarao Khambampati, David Joslin and Austin Tate) led to many clarifications and terminology changes (the notion of auxiliary constraints arose at this time).

Involvement in emerging standards for planning in the military, process modelling, project management, etc., led to the description of <I-N-OVA> as a wholly constraint based ontology that could underpin a strong and simple representation for describing behaviour and activity of all types. Its power was validated by a study of 29 different representations of plans and processes from a wide range of fields as part of the background studies for the creation of the NIST process Specification Language (PSL). <I-N-OVA> came out as having maximum coverage of the detailed requirements list for PSL of all the representations studies. See <http://www.nist.gov/psl/> or Schlenoff et. al. (1999). Indeed those elements it did not cover were related to items to be handled by plug-ins to <I-N-OVA> and deliberately not committed to within <I-N-OVA> itself. A description of the relationships between these various standards efforts and the role <I-N-OVA> played in this is in Tate (1998).

I-X was conceived of in the late 1990s as a way to draw on the best of this earlier work, while making it much more generic and suitable to many kinds of product synthesis or modification tasks. <I-N-CA> is an even more flexible and general purpose constraint-based representation (compared to <I-N-OVA which is a specialisation) of any synthesised artifact. The need for <I-N-CA> emerged during the discussions for the design of I-X and I-Core.

## 5 Summary

I-X draws on the core aspects of work over a 20 year period on O-Plan which provided a flexible component architecture and was an early example of a number of other similar approaches (Blackboard Architectures, NASREM and DICAM).

I-X simplifies and makes more generic the component boundaries and naming conventions used to make the concepts more re-usable for a range of synthesis tasks.

I-X is based on the <I-N-CA> constraint ontology - a powerful and extremely flexible representation of the products of the synthesis process that an I-X system supports. This represents a product as a set of constraints on the space of

all possible products within the model of the domain that the I-X system has. This ontology relates well to emerging standards for process representation and interchange (e.g. in PIF, NIST PSL, DARPA SPAR).

## 6 Acknowledgements

Thanks to my co-workers on the O-Plan and I-X projects at Edinburgh – especially Jeff Dalton, Robert Inder and John Levine for recent discussions on the topic of this paper.

This work is supported via the O-Plan and I-X projects which are sponsored by the Defense Advanced Research Projects Agency (DARPA) and US Air Force Research Laboratory Command and Control Directorate under grant number F30602-99-1-0024 and the UK Defence Evaluation Research Agency (DERA) under the I-Con project.

This work is supported partially under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

The U.S. Government, DERA and the University of Edinburgh are authorised to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of other parties.

## 7 References

- [1] Albus, J.S., McCain, H.G., and Lumia, R. (1987) "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NBS Technical Note 1235, National Bureau of Standards, Gaithersburg, MD, USA.
- [2] Fraser, J. and Tate, A. (1995) "The Enterprise Tool Set -- An Open Enterprise Architecture", Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.
- [3] Hayes-Roth, B. and Hayes-Roth, F. (1979) "A Cognitive Model of Planning", *Cognitive Science*, 1979, pp. 275-310.
- [4] Schlenoff, C., Gruninger, M., Tissot, F., Valois, L., and Lubell, J. (1999) "[The Process Specification Language \(PSL\): Overview and Version 1.0 Specification](#)", NIST Internal Report (NISTIR) 6459, National Institute of Standards and Technology, Gaithersburg, MD, USA
- [5] Tate, A. (1996) "The <I-N-OVA> Constraint Model of Plans", Proceedings of the Third International Conference on Artificial Intelligence Planning Systems, (ed. Drabble, B.), pp. 221-228, Edinburgh, UK, May 1996, AAAI Press.
- [6] Tate, A. (1998) "Roots of SPAR", in "Special Issue on Ontologies", *Knowledge Engineering Review*, Vol.13(1), March, 1998, Cambridge University Press.
- [7] Tate, A. (2000a) "<I-N-OVA> and <I-N-CA> - Representing Plans and other Synthesized Artifacts as a Set of Constraints", AAAI-2000 Workshop on Representational Issues for Real-World Planning Systems, at the National Conference of the American Association of Artificial Intelligence (AAAI-2000), Austin, Texas, USA, August 2000.
- [8] Tate, A. (2000b) "Intelligible AI Planning", in Proceedings of the Twentieth British Computer Society Special Group on Expert Systems International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, UK, December 2000.
- [9] Tate, A., Dalton, J. and Levine, J. (1998) "Generation of Multiple Qualitatively Different Plan Options", Fourth International Conference on AI Planning Systems (AIPS-98), Pittsburgh, PA, USA, June 1998.

- [10] Tate, A., Dalton, J. and Levine, J. (2000a) "O-Plan: a Web-based AI Planning Agent", AAAI-2000 Intelligent Systems Demonstrator, in Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI-2000), Austin, Texas, USA, August 2000.
- [11] Tate, A., Dalton, J., Jarvis, P. and Levine, J. (2000b) "Using AI Planning Technology for Army Small Unit Operations", Poster Paper in the Proceedings of the Artificial Intelligence Planning and Scheduling Systems Conference (AIPS-2000), Breckenridge, Colorado, USA, April 2000.
- [12] Tate, A., Levine, J., Dalton, J. and Nixon, A. (2001) "Task Achieving Agents on the World Wide Web", in "Creating the Semantic Web", Fensel, D., Hendler, J., Liebermann, H. and Wahlster, W. (eds.), MIT Press, 2001.
- [13] Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998) "The Enterprise Ontology", in "Special Issue on Ontologies", Knowledge Engineering Review, Vol.13(1), March, 1998, Cambridge University Press.