Trinity

Modelling methods to support multi-actor problem solving

Henk B. Diepenmaat1,
Lars van Lierop1,
Chris Bruijnes2

1TNO Institute of Environmental Sciences, Energy Research and Process Innovation, Department of Environmental Management,
P.O. Box 342, 7300 AH Apeldoorn, The Netherlands
Phone +31 55 549 3742 Fax +31 55 549 3390 E-mail h.b.diepenmaat@mep.tno.nl

2Ministry of Housing, Spatial Planning and the Environment,
Directorate for Air and Energy
P.O. Box 30945, 2500 GX The Hague, The Netherlands

Abstract
The Trinity approach is an attempt to provide methodological support for knowledge brokerage activities. Knowledge brokers are a type of knowledge managers dedicated to problem solving processes in knowledge-intensive multi-actor problem contexts. Trinity was originally designed for environmental problems. The approach however is generic, and may be used to support multi-actor problem solving processes in other domains as well.

In this paper, some of the foundations of Trinity are sketched. Parts of the Trinity modelling language are presented. The use of Trinity is illustrated by means of parts of a small case study, concerning reduction of the emission of air pollution, that was conducted for the Dutch government. The paper ends with a short discussion of Trinity from a knowledge management point of view.

Introduction
An environmental problem solver in general is confronted with a complex situation of interacting, and in many cases even conflicting, (intentional) activities: many parties can be distinguished, the problem context is multi-actor. Methodological support for problem solving processes in such problem contexts is scanty. The Trinity approach is an attempt to help fill in this methodological gap.

The philosophical, theoretical, methodical and practical aspects of the Trinity approach will be presented in depth in (Diepenmaat 1997).

In this paper first the Trinity interpretations of multi-actor processes, knowledge and problem solving are presented in short. After that, parts of the Trinity modelling language are presented. The use of Trinity is illustrated by means of parts of a case study, concerning reduction of the emission of air pollution, that was conducted for the Dutch government.

Multi-actor problem solving
Typically, in environmental problem solving processes a large number of actors is involved. These actors are either part of the problem context as such, or contribute knowledge to the problem solving process. They belong to many different societal groups (e.g. consumers, producers, governments) and scientific disciplines (e.g. natural sciences, technologies, human sciences). Apart from the number of actors involved, another important characteristic is the complexity of the points of view that motivate and guide the actions of these actors.

In figure 1, both attributes (number of actors involved and complexity of points of view) are used to construct a typology of environmental problem situations that focuses on knowledge distribution. The letters A-D represent typical problem situations. The words in italics refer to typical problem solvers for these situations. Problem types and problem solvers are explained below.
A-type problems in principle are easy problems. Only one point of view (perspective) is relevant for solving the problem. Moreover, this point of view is simple.

In B-type problem situations, several simple points of view must be taken into account. This introduces a need for coherence and co-ordination. A typical problem solver is a general manager.

C-type problem situations require availability of a specialist. Note that, although using the solution of a specialist may be easy, actually understanding his/her reasoning processes may turn out to be difficult.

D-type problem situations are worst case: they are characterised by a lack of coherence and organisation in a complex, multi-perspective interdependent context. Consequently, the key to a solution is a form of coherence between points of view, in which the original problem is solved, and no new insurmountable problems are introduced. A typical problem solver in D-type situations we call a knowledge broker. From figure 1 it follows that in the ideal situation a knowledge broker should possess far more knowledge than professional specialists or co-ordinators. This requirement of course cannot be fulfilled. Knowledge brokers operate at the “difficult” side of the cognitive threshold. The cognitive threshold is an intuitive notion that suggests an upper level with respect to the amount of knowledge that an individual may comprehend. The only way out here is to either use more abstract interpretations of the knowledge involved, or to neglect parts all together. Clearly, knowledge brokerage activities are demanding.

Typical examples are policy makers and managers of complex projects.

A knowledge broker is working in a problem context, involving many different, complex points of view. As a result, a knowledge broker is focused on knowledge distributions and the processes that (may) change them. These points of view can have for example a personal, disciplinary and/or cultural background.

A commonly used strategy to cope with D-type situations is to train people as generalists. This at least covers the requirement that a D-type problem solver preferably should possess advance knowledge with respect to the problem context. This requirement is implied by the fact that in the D-region of figure 1 complexity (the horizontal axis) is large: many difficult points of view are relevant.

In addition to possessing advance knowledge, however, the ability to obtain, communicate and integrate knowledge within reasonable limits of time and effort is perhaps an even more crucial requirement for knowledge brokerage activities. This requirement may be summarised as the necessity to possess knowledge management knowledge and skills. It is implied by the fact that the number of intentional actors (each of them acting according to their private knowledge, see further on) is large (the vertical axis).

A third important characteristic of knowledge brokerage activities is that a knowledge broker should avoid to insert his/her own point of view (an attitude not atypical for C-type domain specialists), or to direct too dominantly (an attitude not atypical for B-type general managers). Rather, focus should be on obtaining an understanding of points of view, and bringing them into coherence. This is where the notion of modelling may be of great help.

A knowledge broker therefore can be characterised as:
- a domain knowledge generalist (as the complexity, the horizontal axis, is high);
- a knowledge management specialist (as the number of intentional actors, the vertical axis, is high); and
- oriented at bringing into coherence knowledge, possessed by other actors involved.

An overview of the relation between problem types, methodological considerations and typical problem solvers is presented in table 1.

![Figure 1: Trinity is directed at supporting knowledge brokers in D-type problem solving.](image)
Table 1 Problem types, methodological principles and typical problem solvers.

<table>
<thead>
<tr>
<th>Type</th>
<th>Methodological principles</th>
<th>Typical problem solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-type</td>
<td>Trivial</td>
<td>Trivial</td>
</tr>
<tr>
<td>B-type</td>
<td>Co-ordination</td>
<td>General manager / Co-ordinator</td>
</tr>
<tr>
<td>C-type</td>
<td>Domain specific</td>
<td>Specialist</td>
</tr>
<tr>
<td>D-type</td>
<td>Knowledge modelling, knowledge communication and knowledge integration methods</td>
<td>Knowledge Broker</td>
</tr>
</tbody>
</table>

The Trinity world view: interpretations of problem solving and knowledge

According to the Trinity world view, any phenomenon can be described (and ultimately modelled, see further on) as a systemic construct of activities. Only two types of activities are distinguished within Trinity: intentional activities and autonomous activities.

According to the Trinity approach, an intentional activity consists of the following stages:

- **stage 1**: acknowledge situation of concern
- **stage 2**: construct perspective
  - **stage 2a**: analyse situation "as is"
  - **stage 2b**: synthesise script (a plan for taking action)
  - **stage 2c**: predict situation "to be"
- **stage 3**: implement script (act in environment)
- **stage 4**: evaluate situation "to be"

Figure 2: A stage-based model of an intentional activity

An example is the activity of buying a bottle of milk in a supermarket (which would be an A-type activity). When opening the refrigerator, one acknowledges that there is no milk (stage 1); a perspective is constructed that might involve visiting the supermarket (stage 2), the script part is actually executed (stage 3), and the result is that there is milk again in the refrigerator (stage 4). However, the same cycle can be used to model far more complex activities (like realising a sustainable society).

Stage 2 results in a perspective: a body of knowledge, possessed by an intentional actor, that motivates and guides his/her action (that makes this action intentional). This implies that, in terms of Trinity, knowledge consists of perspectives, and knowledge processes operate on and result in perspectives.

If stage 2 is non-trivial (requires substantial cognitive effort; an appropriate perspective is not readily available), it corresponds with a problem solving process. A problem manifests itself (is acknowledged, or rather: emerges) as a mismatch between the environment and the intentions of an actor. In terms of Trinity, problem solving is perspective construction.

The other type of activities that is distinguished in Trinity are autonomous activities. Autonomous activities are activities that take place "by themselves": an intentional actor or a guiding and motivating perspective is not required. An example is the emission of methane from a swamp, or the burning of a fire (lighting a fire typically is an intentional activity).

Also in case of autonomous activities a situation "as is" and a situation "to be" are distinguished, as well as a process that realises the transition. However, stages like "acknowledge", "construct perspective", "implement" and "evaluate" hardly are appropriate. Rather, the "cycle" simply is:

- **stage 1**: a situation "as is" manifests itself
- **stage 2**: (therefore) a process happens
- **stage 3**: which results in a situation "to be"

Figure 3: A stage-based model of an autonomous activity

Cycles of (intentional and autonomous) activities can be used in combination. For example, they can be distinguished recursively (at several levels of detail), sequentially (a spiral), concurrently, et cetera. From a Trinity point of view, real world phenomena are (or at least can be interpreted, hence modelled as) systemic, self-contained constructs consisting of activities (figure 4). As intentional activities require knowledge (they are guided and motivated by perspectives), this idea can be used as a basis for model-based support for knowledge management activities in D-type problem solving processes. This will be elaborated below.
Figure 4: Real world phenomena can be interpreted as systemic constructs of intentional and autonomous activities.

From a modelling point of view, autonomous activities are easy to model: as is states are turned into to be states by means of processes. Intentional activities are more difficult: they require some acknowledgment, the possession of a perspective, the implementation of a script and an evaluation. However, the net result also is that an as is state is turned into a to be state by means of a process (an action).

The Trinity principle

The Trinity principle states that real world phenomena take place in (a combination of) three different domains. These three domains are: the knowledge domain, the physical domain and the communication domain.

The knowledge domain deals with activities that change the "world of perspectives" that we possess and may use to act intentionally. Examples of parts of this world are the "open wine bottle" perspective, the "sharpen pencil" perspective.

The communication domain deals with activities that change the "world of representations". Examples of parts of this world are a letter, a book, a speech, a model.

The physical domain covers the "world of things", i.e. the objects that, at least in principle, are observable (sensory perceptible) by anybody, and that are not part of the communication domain.

In each of these three domains both intentional and autonomous activities may take place. For example, forgetting is an autonomous activity in the knowledge domain; taking a course is an intentional activity in the knowledge domain. Burning a book in a cold winter is an intentional activity in the physical domain, whereas burning the same book for religious reasons would be a communication domain intentional activity.

This implies that in each of the three domains both domain states and processes exist. States are the in- and outputs of processes. Processes transform states: they change a domain.

According to the Trinity point of view, states and processes do not exist in some absolute sense. Physical domain examples are: the presence of a Deux-Chevaux is a state, but at the same time this beautiful car is likely to corrode, which is a process; the pencil in my hand is a state, but at the same time the molecules constituting this pencil vibrate; the rivers in the Netherlands are states from a geographical point of view, but they flow at the same time, and from a historical perspective they even move. An example in the knowledge domain is: universities possess the knowledge to perform research activities (a state), but at the same time this knowledge continuously changes (a process). A communication domain example is: a library consists of books (a state), but the books actually present in the library change over time (a process).

Therefore, we define a state as something that is static with respect to the (intentional or autonomous) processes that one is interested in. Conversely, a process is something that changes (or results in) a state that one is interested in. From a philosophical point of view, this is a combination of a relativistic and a pragmatic interpretation (something is the action potential it offers to someone).

This relativistic nature of states and processes enables to design abstraction and specification procedures in the Trinity modelling language.

The Trinity modelling language

The operational manifestation of Trinity is that it provides a modelling language that supports the perspective construction process in D-type problem contexts. The language enables a knowledge broker to represent, adapt, revise and discuss D-type perspectives. From a strict point of view, the "sharpen pencil" perspective and the "realise sustainable society" perspective are quite similar in that they both consist of an "as is" part, a script part and a "to be" part. The latter however is a D-type perspective: the "as is" part, the script part and the "to be" part contains references to intentional activities of many other actors (the problem is D-type). Note that this implies a recursion (a shift in level of reflection): the knowledge broker is engaged in an intentional activity, directed at changing a domain of operation, but the domain of operation of this broker contains many intentional activities on its turn, who are dealing with their own domain of operation. In sharpening a pencil other actors are not likely to be relevant: the domain of operation is physical. In terms of the Trinity principle: D-type activities take place in a problem context in which many intentional knowledge domain activities take place at the domain level.

From the paper so far it follows that the Trinity modelling language should enable to represent systemic constructs of intentional and autonomous activities in three domains.

Model primitives
We will use an ellipse to refer to a process (in any of the three domains); a hexagon to refer to a perspective (a knowledge domain state), a rounded box to refer to a communication domain state, and a rectangle to refer to a physical state. Figure 5 presents examples.

Figure 5: Reference model primitives.

The examples illustrate that the model primitives can be numbered, and are provided with a name that should enable to understand the semantic correspondence (i.e. to relate the symbol to its domain referent). The name explicitly refers. For this reason, they are called reference model primitives. A process (ellipse) typically has an "active" name like "mows lawn" or "breaks". A communication domain state (rounded box) typically has a name that refers to a physical communication means, like "book", "report", "speech", "library". A physical state (rectangle) typically has a name like "tower", "bridge", "molecule". A hexagon (perspective) typically is named after the purposeful action that it may motivate and guide: for example the "research performer" perspective consists of the knowledge required to perform research; the "lawn mower" perspective consists of the knowledge required to mow a lawn; the "air traveller" perspective consists of the knowledge required to be able to travel by air. Note that these names correspond with the name of the actor who possesses the perspective and is able to act according to it. In order to facilitate semantic correspondence (to relate a symbol with its referent), model primitives can be further described in "natural language", by means of a textual attachment. The number in combination with the name provides a reference to this textual attachment. Another way to elaborate a model primitive is to detail it on a lower systemic level by means of specifications (see further on).

Modelling intentional and autonomous processes
The central model primitive of an intentional activity is the perspective, as the perspective is linked with all three other stages (1, 3 and 4). Figure 6 represents an example of the way in which an intentional activity is modelled in the Trinity language.

Figure 6: A physical domain intentional activity.

The reference model primitives are connected by means of causation arrows (somehow, the initial reference model primitive causes, is responsible for, the final reference model primitive). Arrows do not explicitly refer: they derive meaning from the way in which they are connected. Note that the six corners of the hexagon correspond with the six stages (3 sub-stages included) of an intentional activity. Knowledge domain and communication domain intentional activities are modelled in the same way. The environment as is and to be become hexagons and rounded boxes respectively. In case of knowledge domain as is and to be states, arrows depart and enter from the middles of the sides, in order to express that they are part of the domain of operation, rather than core part of an intentional activity.

Autonomous activities are more simple to represent: the central hexagon is not required. Figure 7 presents a physical domain example. The example in addition makes clear that several arrows entering or leaving a reference model primitive should be interpreted as a logical "and".

Figure 7: A physical domain autonomous activity.

Adapting the models
The relativistic interpretation of states and processes allowed for the construction of a library of extension,
restriction, abstraction and specification procedures that enable to change the scope of a model, or to change level of detail (or use multiple levels of detail). This library of elementary modelling steps enabled us to derive a large family of modelling strategies (typical sequences of modelling steps), that provide a way of thinking about modelling processes that is more in terms of overall goals. Note that the notions of modelling steps and modelling strategies as presented above merely utilise the systemic nature of intentional and autonomous activities. Below first an overview of modelling strategies is provided (figure 8, for an elaboration see Diepenmaat, 1997). After that two examples are presented (figure 9a and 9b).

**Figure 8: Modelling strategies.**

Figure 9a presents a transformation strategy. The scope remains the same, but the level of detail changes.

**Figure 9a: A transformation strategy.**

Figure 9b presents an example that mixes scope and level of detail variations. The strategy is called “bird’s eye” as
level of detail is traded of for scope. The number of model primitives in the initial and the final model however is the same (for this reason it also is a balancing strategy).

**A case study:** reducing the emissions of Volatile Organic Compounds

In the period May 1986 to April 1988 a policy was formulated to effect a reduction of Volatile Organic Compounds (VOCs) into the air by industry, small businesses and household in the Netherlands. This programme is based on an agreement between public authorities, provinces and municipalities and trade and industry to effect the reduction of Volatile Organic Compounds (VOCs) into the air by industry, small businesses and households by 63 % in the year 2000 (compared to 1981). The main goal of this reduction is to reduce the effects of ozone in the lower layer of the atmosphere (this so called “low ozone” causes damage to materials, plants and crops, and influences public health). The present strategy consists of reduction plans per sector. Such a reduction plan typically indicates the types of measurements that should be taken in order to reduce the VOC emission. The implementation of the strategy is in an advanced phase. Many sectors are on schedule (the overall programme is successful), others are delayed, and in some sectors only a small improvement, if at all, is noticeable.

A *Trinity* analysis clearly indicated that one of the important reasons of the delays is an increasing mismatch between the predominantly technical nature of the VOC programme and the far more complex nature of the required processes of change in industrial sectors. In terms of the typology presented earlier in this paper: implementing reduction measurements are too often interpreted to be C-type, while a more careful analysis reveals their true D-type nature. Typically, many different actors are involved on regional, national, international and even global scales. *Trinity* modelling processes started with drawing up lists of these actors (sources are interviews, documents, workshops, et cetera), and after that bringing them into a coherent “whole” by means of models.

A simplified example of a *Trinity* model of a VOC perspective, dealing with the process of printing packing materials, is presented in figure 10. It is presented merely as an example, and not explained any further (for an elaboration see Diepenmaat, 1997). In the near future, *Trinity* will be used to support the process of finding feasible alternatives to implement emission reduction measurements in complex situations. Note that *Trinity* supports perspective construction, and does not generate a solution (a decision). In this respect, *Trinity* falls out of the scope of decision making (although the alternative perspectives to decide on will be more clear).

**Discussion and conclusions**

It was rather surprising that in many cases the step from actor lists towards models proved to be very difficult. At first we seriously started to question the *Trinity* methodology. However, after more detailed analysis the reason for this turned out to be quite different: documents and speech are far too permissive to result in a clear insight in change processes in complex networks. This insight is substantiated by experiments directed at strategic environmental conferences (where up to 160 actors could be distinguished, see also Diepenmaat, 1997). Using *Trinity* results in a far more clear picture of both actors in isolation (their knowledge, their actions, their possibilities to change) and actors as a part of the network, and this in the actual situation (as is), the process of change (the script) and the expected to be situation.

Although *Trinity* is “in statu nascendi”, the new users of this moment state that, after a first period of learning the conceptual framework, *Trinity* rigidly changes their way of looking at and operating in D-type contexts. We suggest that the main reason for this is the fact that *Trinity* requires an explicit reflection process with respect to intentions, knowledge and actions of actors, an issue that in normal practice is given too little attention.

*Trinity* models explicitly refer to perspectives in as is, to be and script models. This implies that they support knowledge inventories, knowledge requirements in improved situations, as well as knowledge processes (diffusion, transfer et cetera). In this sense, *Trinity* supports knowledge management in highly dynamic D-type contexts.

**Acknowledgements**

The authors appreciate the contributions of Jacqueline Cramer, Wim Harder and Bob Wielinga to the development of the *Trinity* approach.

**References**

*Trinity* is built upon many different sources and concepts from many different disciplines. It is described in depth (including these references) in a dissertation that at this moment exists in a concept version only and will be published in 1997: