Intelligent Manager for Comprehensive Design
IMCOD

Michael M. Richter, Bernd Bachmann, Ansgar Bernardi,
Gabriele Schmidt and Anna–Maria Schoeller

German Research Center for Artificial Intelligence — DFKI GmbH,
P. O. Box 2080,
D–67608 Kaiserslautern,
Germany,
{sek-mmr,bachmann,bernardi,schmidt,schoeller}@dfki.uni-kl.de

Abstract

The complexity of modern product design requires a flexible organization of the development process. Arising tasks are delegated to specialized experts, their contributions are combined into the final design. All this poses high requirements on the product manager who is responsible for the complete development process. The product manager must manage the process, i.e. he delegates tasks, communicates with the experts and makes decisions. IMCOD supports this activities. Based on process models and descriptions of all participants the system guides the process, provides the necessary communication interfaces and performs multi–criteria decisions. To do this IMCOD employs a scenario of individual participants contributing to the common goal under central management. IMCOD is applied to an example from the domain of design of pressure tanks made of composite materials.

Contents

1 Introduction 2
1.1 IMCOD's Motivation by the Application 2
1.2 Contents of this Report 2

2 The Abstract Scenario 3
2.1 Participants 4
2.1.1 The manager 4
2.1.2 The expert Exp_1,..., Exp_n 5
2.1.3 Initial designer 6
2.1.4 Designer: 6
2.1.5 External participants: 6
2.2 The Network 6
2.3 The Control Structure 7

3 The Application Domain of Pressure Tanks 8
3.1 The Situation Presented by the IVW GmbH 8
3.2 Customer Requirements 9
3.3 Selection of Relevant Norms 10

4 Instantiating the Abstract Scenario by the Example 12
4.1 Participants 12
4.2 The Tasks of IMCOD 12

5 The Construction Process for the Concrete Example 15
5.1 Formalizing the Requirements 15
5.1.1 Formalization templates 16
5.1.2 The formalized requirements: The design specification 16
5.2 Initial Design Suggestions 17
5.3 Communication with the Participants 18
5.3.1 Modeling the ontologies 18
5.3.2 Translation rules 21
5.3.3 Modeling the interface 21
5.3.4 Applying the model of the participant 22
5.4 Evaluation Dialog 23
5.5 Multi–Criteria Decision 23
5.6 The Result: The Selected Design 24

6 Discussion and Conclusion 25

1 Introduction

1.1 IMCOD's Motivation by the Application

Modern industry faces some challenging requirements: Modern products must offer ever–increasing quality at market–conforming prices. Increasing numbers of different product variants and the constant request for shorter response times and the shortest possible time–to–market ask for new flexibilities in every facet of the production process.
Additional constraints arise out of the political field: Growing environment–awareness leads to new laws and tend to underline a companies responsibility for the whole life–cycle of their products. Consequently problems like recycling of the product and environmental impact of the production process have to be taken into account.
One approach to reach progress under these circumstances is an organizational change: In an idealized scenario one (human) *product manager* is responsible for every aspect of a product, starting at the first conception and ending at the recycling.
The design of a product plays a central role in this scenario. Although the design accounts only for a relatively small part of product costs, the decisions which have been fixed during the design period are responsible for more than 80 % of the total cost of a product. This holds also for other problems mentioned above: Recycling problems are created or solved mainly during the design phase, a successful design is responsible for a short time–to–market and so on.
Due to this observation any approach to support the modern product generation process has to take into account the design phase of the product. The main problem can be described as a problem of information transfer: The product manager needs any relevant information about every aspect of the product's life–cycle at the design phase.
IMCOD supports this aspect of product development. The IMCOD system shall monitor the design process, comparing it to predefined process models, thus assuring the realization of well–defined procedures. Just at the right moment during design the system shall automatically communicate with relevant experts (systems or human), thus enabling optimal use of expertise. Integration of diverging terminologies and differing preferences of various information sources together with automatic calculation of multi–criteria decision models shall lead to a homogeneous presentation of relevant information to the product manager. The documentation of design decisions with reasons and consequences forms a basis for quality assurance and allows a better understanding of design results at a later time.
In summary, IMCOD supports a product manager's task by providing optimal access to several information sources.

1.2 Contents of this Report

In this report, we will demonstrate the functionality of the IMCOD system in the context of a larger example: the design of a pressure tank for a truck's brake system using composite materials. This example was provided by the IVW GmbH (Institut für Verbundwerkstoffe), Kaiserslautern.
In order to structure the description, we will introduce an abstract scenario which will then be instantiated by the example. First, we will repeat here shortly some major aspects of

IMCOD. In order to avoid a terminological confusion, we distinguish two constructive aspects:

- The construction to be carried out (in our example: pressure tanks) will be called the *design*.
- The task to be performed by the IMCOD system will be called the *construction process*. The goal of the construction process is the development of the design.

The designs are not always completed and then called partial designs. To perform the construction process, the IMCOD system acts as a planner to whom the usual terms of production planning apply. We consider planning as decision making and have the following abstract process in mind.
The planner (i.e. the decision maker) has the task to transform a given situation into a new one which satisfies certain conditions and approximates certain optimality criteria by carrying out certain activities. The planner analyses the situation and investigates available actions. The resulting situations are again analyzed and in particular evaluated. These steps are iterated and withdrawn if necessary which may lead to some backtracking steps.
In ordinary planning, the planner performs his task without any help from other persons or systems. In IMCOD, the planner is supported by several other figures in the scenario. The main point is that this scenario has a clear structure in which everybody has a fixed role. We now distinguish the management aspect of the construction process (delegation of tasks, communication, decision making) form the remaining construction activities. The latter are performed by the other participants while the IMCOD system concentrates on the management aspect. Thus, IMCOD (dynamically) organizes the global flow of information which is illustrated in the following diagram (see figure 1).
It is the purpose of the example to explain this structure. Therefore, several other aspects and activities occurring in planning systems have been degenerated or completely left out. Of course, in the final IMCOD system many of them will appear again.

2 The Abstract Scenario

In the description of the abstract scenario for IMCOD the participants as well as their knowledge and functionalities are mentioned. Each participant of the scenario is characterized by

- the static knowledge base
- the dynamic knowledge base
- the functionalities

The scenario describes also the connections between the participants and the control structure. On this abstract level, the scheme describes how a design is created in IMCOD.

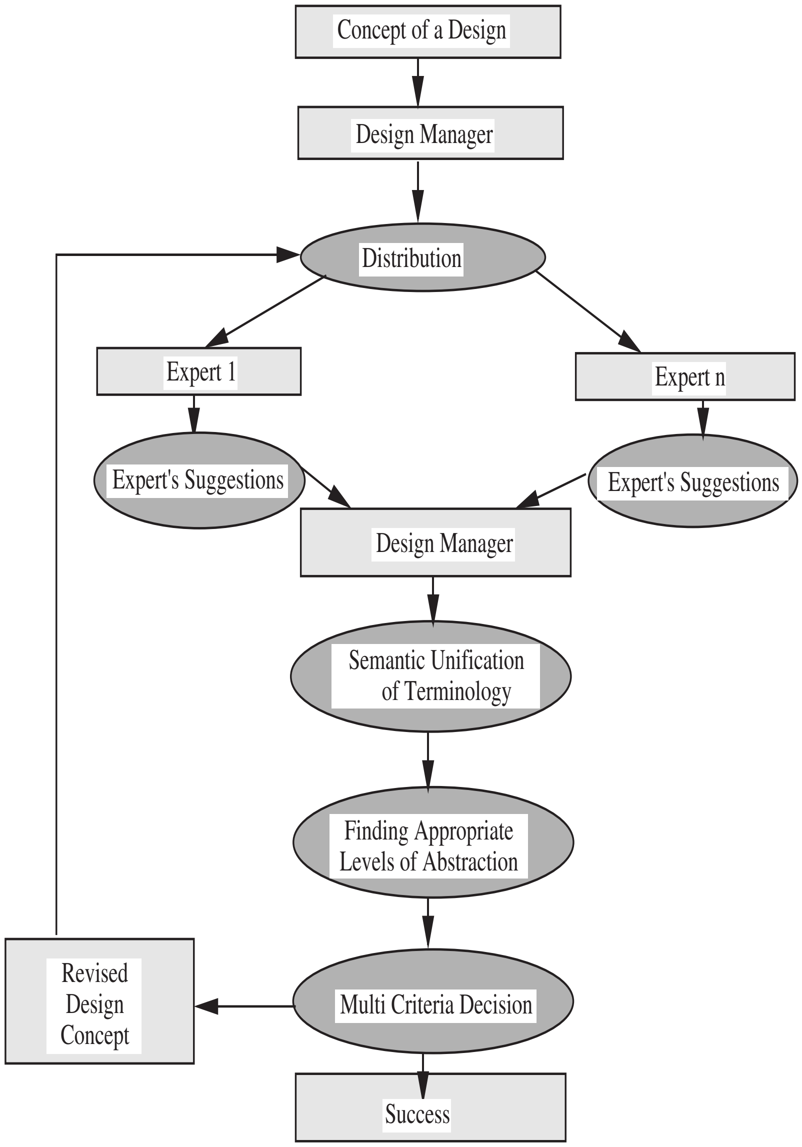

Figure 1: General information flow in IMCOD

2.1 Participants

2.1.1 The manager

Static knowledge base:

- Knowledge on experts
- General knowledge on
  - hard constraints for designs

  - soft constraints for designs (preferences)
- Relationship between preferences
- Control knowledge

Dynamic knowledge base:

- Design specification
- Hard constraints
- Soft constraints
- Partial designs
- Explanations from experts

Functionalities

- Selection of experts
- Requesting experts
- Refusal or acceptance of construction proposals
- Establishment of utility functions
- Requesting designers
- Handling of control

The manager is responsible for all decisions to be made and all actions to be carried out. The IMCOD system is intended to realize important parts of the manager's functionalities, thus supporting the human product manager.

2.1.2 The expert $\mathbf{Exp}_1$,..., $\mathbf{Exp}_n$

Static knowledge base:

- Expert knowledge$_1$,..., expert knowledge$_n$

Dynamic knowledge base:

- Partial designs, constraints

Functionalities

- Rejection of designs
- Utility functions of design
- Explanations

2.1.3 Initial designer

Static knowledge base:

- Design specifications

Dynamic knowledge base:

- Design constraints

Functionality

- Distribution of initial designs (partial)

The generation of initial designs from scratch is a highly creative process. In reality, this task will usually be performed by humans, although systems like databases of previous designs might provide valuable support.

2.1.4 Designer:

Static knowledge base:

- Design know–how

Dynamic knowledge base:

- partial Designs, constraints, specifications

Functionality

- Partial completion of partial designs

2.1.5 External participants:

Functionality

- Providing specifications, constraints, general regulations, etc.

The (initial) designers are special experts who are emphasized because they are absolutely necessary in the scenario. They could also take over tasks of other expert.

2.2 The Network

The network shows that the manager is connected with every other participant, but no two of these other participants have a connection (cf. figure 2). This indicates the basic difference between IMCOD and multi–agent systems.
Along the connections the communication and flow of information takes place. Since the participants have different languages and levels of abstraction interfaces are needed in order to allow communication:
manager — $\boxed{\text{interface}}$ — participant
The main task of the interfaces is to provide a semantic unification.

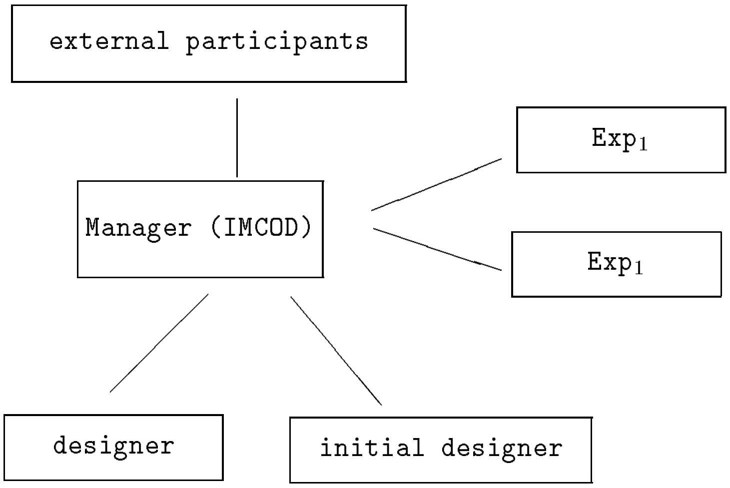

Figure 2: Network of the participants of the abstract scenario

2.3 The Control Structure

Responsible for the control is the manager. As mentioned above this is the main task of the IMCOD system. The control structure will be explained by interpreting the IMCOD flow diagram (see figure 1).

1. External participants —— specification documents: specification, constraints $\longrightarrow$ manager
2. The manager formalizes the documents if not already done.
3. Manager ——— inquiry $\longrightarrow$ initial designer

   The inquiry consists of specifications, related constraints, possible specific questions from the manager.
4. Initial designer——initial designs (suggestions) $\longrightarrow$ manager
5. The manager complements the initial designs and selects experts Exp_{k1}, ... Exp_{kl}.
6. Manager———inquiry $\longrightarrow$ Exp_{kj} $j = 1,\ldots,l$

   Inquiry: partial design, constraints, parts of a design specification (catalogue of questions)
7. Exp_{kj}————Answer $\longrightarrow$ manager

   The answer consists of replies to the questions, preferences of partial designs, explanations.
8. The manager evaluates the answers

9. Manager—order ⟶ designer

   The order consists of partial designs, specific constraints, and a new partial design level.

   The partial design level states how far the design has to be completed.

10. Designer——answer⟶ manager

    The answer consists mainly of a partially completed design.

11. The manager accepts the design or goes to step 3 or step 9

There are basically two loops:

- The external loop: 3) - 11)
- The internal loop: 9) - 11)

Control is performed by the manager: The manager determines

- when to leave the internal loop
- when termination is reached.

For this purpose, the IMCOD system acts as a planner in the usual sense. In particular, the IMCOD system has to administer the partial designs in the dynamical knowledge base, observe dependencies, organize backtracking etc.

3 The Application Domain of Pressure Tanks

3.1 The Situation Presented by the IVW GmbH

The IVW GmbH provides us with the necessary information and knowledge about the design of pressure tanks made of composite materials [10]. The design of pressure tanks as carried out by the IVW GmbH distinguishes two levels of completion (figure 3): the first layout and the constructional layout. The design process begins with the first layout (see left circle of figure 3), where the specifications, the experts' knowledge of design and design databases constitute input data and prerequisites. It is a creative process in which several suggestions for the design are made. If and only if one of these suggestions can fulfill all requirements of the specification the constructional layout (see right circle of figure 3) is started, otherwise, the project is canceled. In the constructional layout, the best suggestion of the first layout is refined.
In our example we concentrate on the first layout. This phase comprises the creative process of development of partial designs, the delegation of evaluation tasks to several experts as well as multi–criteria decisions. Thus, the first layout is well suited to demonstrate IMCOD's functionality.

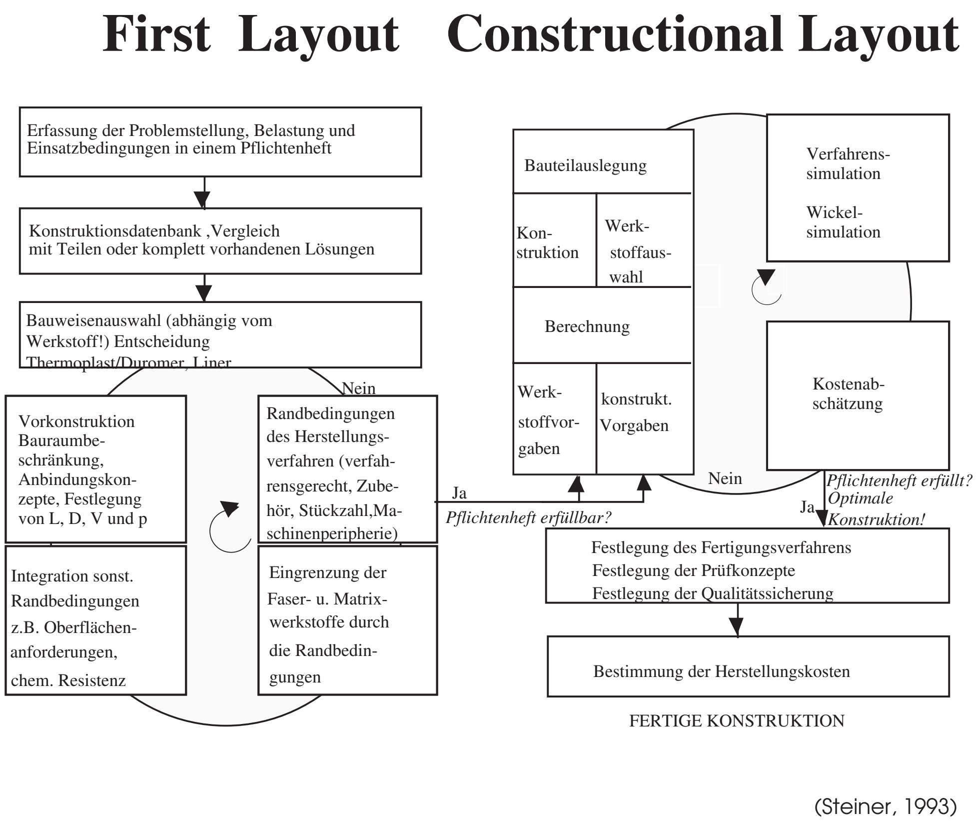

Figure 3: Process and information sources for the design of pressure tanks

3.2 Customer Requirements

Initially the customer states his requirements. They represent the criteria the customer wishes the product to fulfill. These criteria are of different preciseness and of different importance: The customer might state criteria which must be fulfilled, e.g. *pressure = 8 bar*. Any design which does not fulfill these criteria is not valid. He might also state wishes he would like to see fulfilled, e.g. a certain color of the pressure tank. A design which does not fulfill one of these might be valid nevertheless, although another design which fulfills these criteria would be judged better.

Our customer provided the following requirements:

> A pressure tank for a truck's brake system
>
> Requirements for the development of a pressure tank model of composite materials.
>
> volume: 80 Liter
>
> max. diameter: 350 mm
>
> pressure: 8.5 bar while operating
>
> ± 2 bar max. fluctuation

17 bar max. test pressure
The pressure drops off to 0 while the vehicle is not operating.
shape: cylindrical; installed laterally/longitudinally; mechanically attached to the vehicle's body, but removeable for repairs.
No. of units: an estimated requirement of 45,000 units per year
service pipes: 2 at the front, 1 concentric for pressure screw joint/connection, inside thread Rp 3/8; for overpressure-valve and for drainage; the latter manually operable
max. acceleration: frontal collision at 100 km/h
max. endurance: collision from side, e.g. by a motorcycle
durability: against mineral oil, bases, UV rays, paint

Conditions of use The tank should be of light-weight contruction, the durability must be guaranteed and changes in shape should be minimal. The material is only to absorb small amounts of liquids, and it must withstand temperatures ranging from -40^0C to $+120^0$C. It will be utilized on construction sites (quarries) and streets worldwide (climate).

Technical Constraints The tank should not have sharp edges; it should have a clean and smooth surface (visible part). It should be assembled to AD and DIN norms, so that it will pass inspection (TÜV). Furthermore, the quality of the tank should assure DIN ISO 9000 to 9004 norms. For assembly into the vehicles, the tanks must be able to be transported to the automobile manufacturer. All service pipes should be free of wear and tear, the tank itself should be maintenance free and exchangeable which does not mean repairable. If possible, it should be reusable; in any case, the tanks should be recycleable (recycleable material and construction/design).

Economic conditions max. costs: 100 DM/piece
expected profit: 30% per piece
max. investment: 5 mio. DM
depreciation: 5 years
existing equipment: factory building, no machinery

3.3 Selection of Relevant Norms

The design of pressure tanks is governed by a large body of normative texts. Depending on the customer requirements it is possible to select relevant parts of these norms and to exclude irrelevant ones. This process is highly complex due to the legalese language in the normative documents and due to the volume of possibly relevant norms. Figure 4 shows the structure of some normative texts relevant for pressure tanks. Here we find e.g. information about the minimal thickness of the material or formulas to calculate strengths and forces.
Any norm relevant to the design in question can be interpreted as adding some requirements to the initial ones.

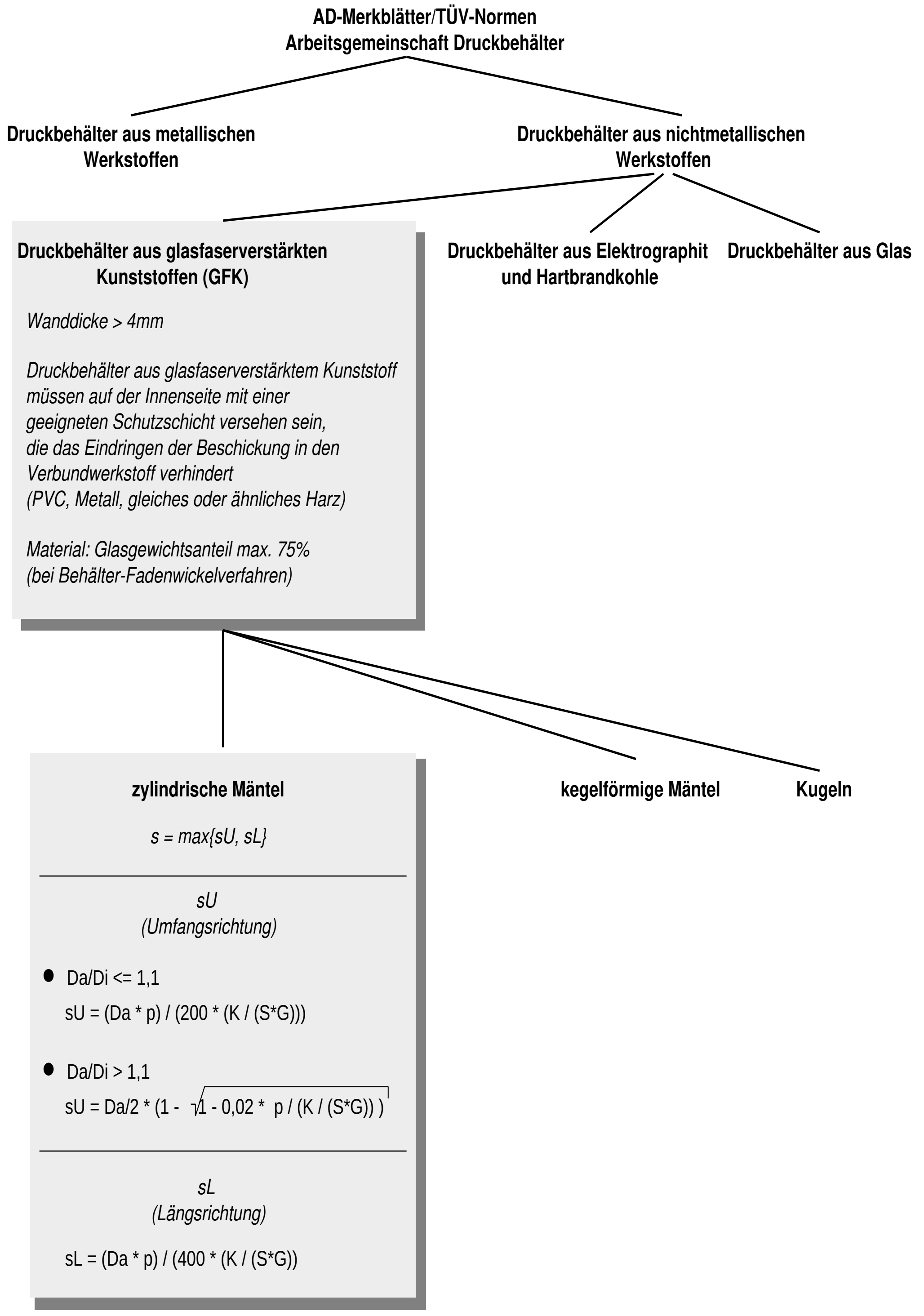

Figure 4: Some relevant norms for pressure tanks

4 Instantiating the Abstract Scenario by the Example

4.1 Participants

First we list the participants involved in the process of pressure tank design.

- In our example we deal with two external participants: the customer and the (documents of the) normative institutions. Both are sources of informal data.

  1. The customer provides the requirements for the design. The contents of the specification has been illustrated above.
  2. Normative institutions determine norms and technical guidelines. In the area of pressure tanks the relevant institutions are the *Technischer Überwachungsverein TÜV* and the *Arbeitsgemeinschaft Druckbehälter AD*. While the former is responsible for testing installations for high pressure and provides relevant security norms, the latter distributes large collections of technical guidelines which guarantee constructions of high quality.

- Several experts calculate values, evaluate the designs and check the fulfillment of the requirements. In our document we refer to the following experts:

  1. The expert for production is competent in questions about the manufacturability of a proposed design. He may also calculate production costs.
  2. The expert for material knows the availability and the technical details of the different materials. Based on this he can judge the suitability of a material for a particular purpose and estimate the costs of recycling a material.
  3. The expert for geometry and tolerances performs the necessary geometric reasoning. He calculates values, e.g. the interdependent values for length, diameter and volume of the pressure tank, determines the overall form (cylindrical or spherical?) and argues about the surface finish.
  4. An expert for quality assurance is responsible for statements about the testability of the design and the fulfillment of documentary requirements.

- The initial designer provides suggestions for partial designs. In our example, the design suggestions are provided by a human participant.

- The designer mentioned in the abstract scenario only appears in the constructional layout phase of the process. Thus, he is not of interest in our example.

4.2 The Tasks of IMCOD

To describe how the IMCOD system should proceed in order to solve the problem of designing a pressure tank we transformed figure 3 which we obtained from the experts at IVW GmbH into our terminology. For the description we selected a KADS inference structure [11]. The inference structure of the design process is developed with the help of CoMo–Kit [7]. The result is presented in figure 5. The terminology of the model, i.e., the

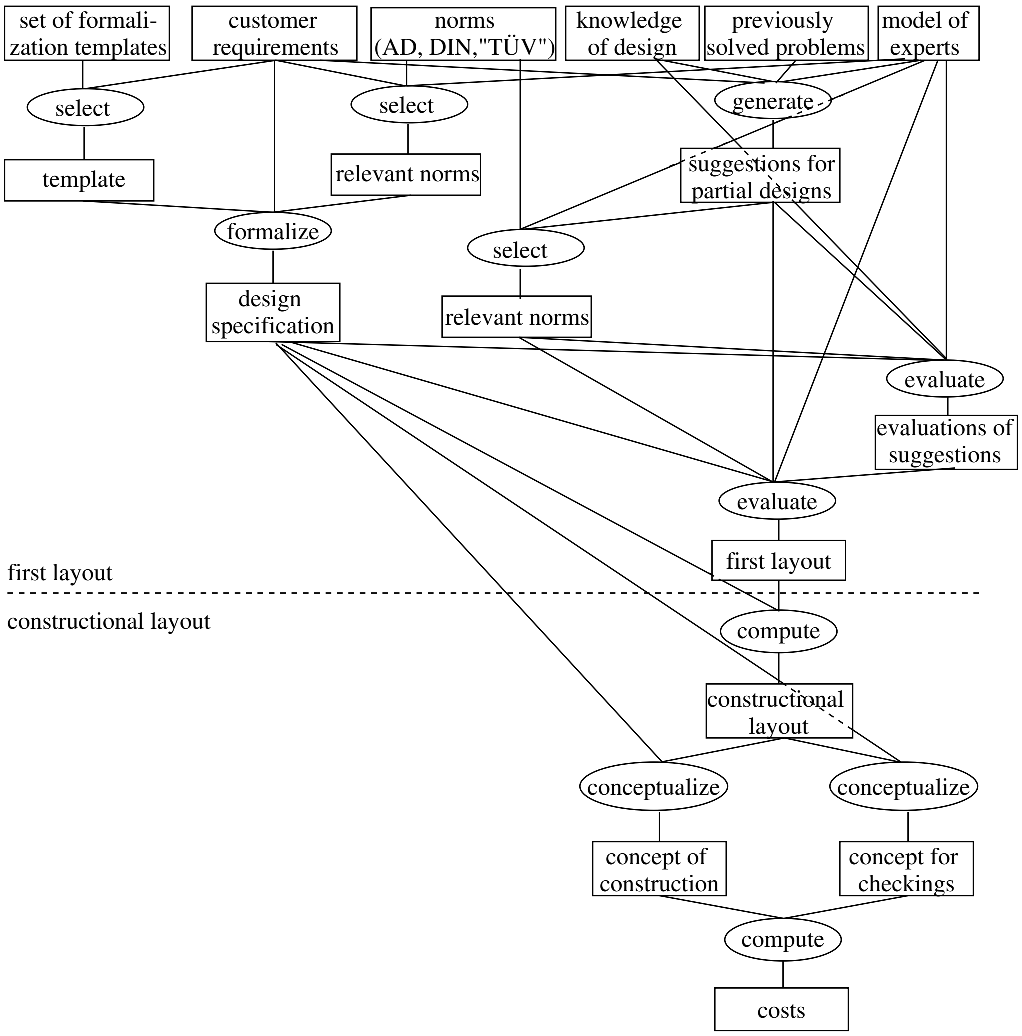

Figure 5: Inference structure of the construction process

names of the boxes and ellipses, is left in the terminology of the application domain, so that the experts were able to evaluate our model.

The inference structure shows the possible information flow if the figure 5 is read from top to bottom[1]. Thereby, the boxes at the top of the figure — which describe roles of domain knowledge according to KADS — can be interpreted as initial knowledge and input data fed to the system.

The other boxes describe roles of domain knowledge — or, from a point of view of IMCOD, roles of data — which must be provided or are required by the participants of the scenario. In other words, the boxes illustrate the input/output data for/of activities performed

[1]An inference structure strictly developed according to the KADS methodology does not show any control information. Therefore, there are only lines and no arrows used in KADS inference structures.

by the participants of our example. The activities are drawn as ellipses. The IMCOD system only covers some of those activities. The remaining tasks are performed by other participants. Their details are not of interest for the development of IMCOD, i.e. the other participants are treated as black boxes.
From the point of view of IMCOD the inference structure must be enhanced by the knowledge which participant performes which activities, which participant needs which data and which participant can provide which data[2]. Since in our example we concentrate on the first layout of the design process this part of the inference structure is enhanced by the required additional knowledge and presented in figure 6.

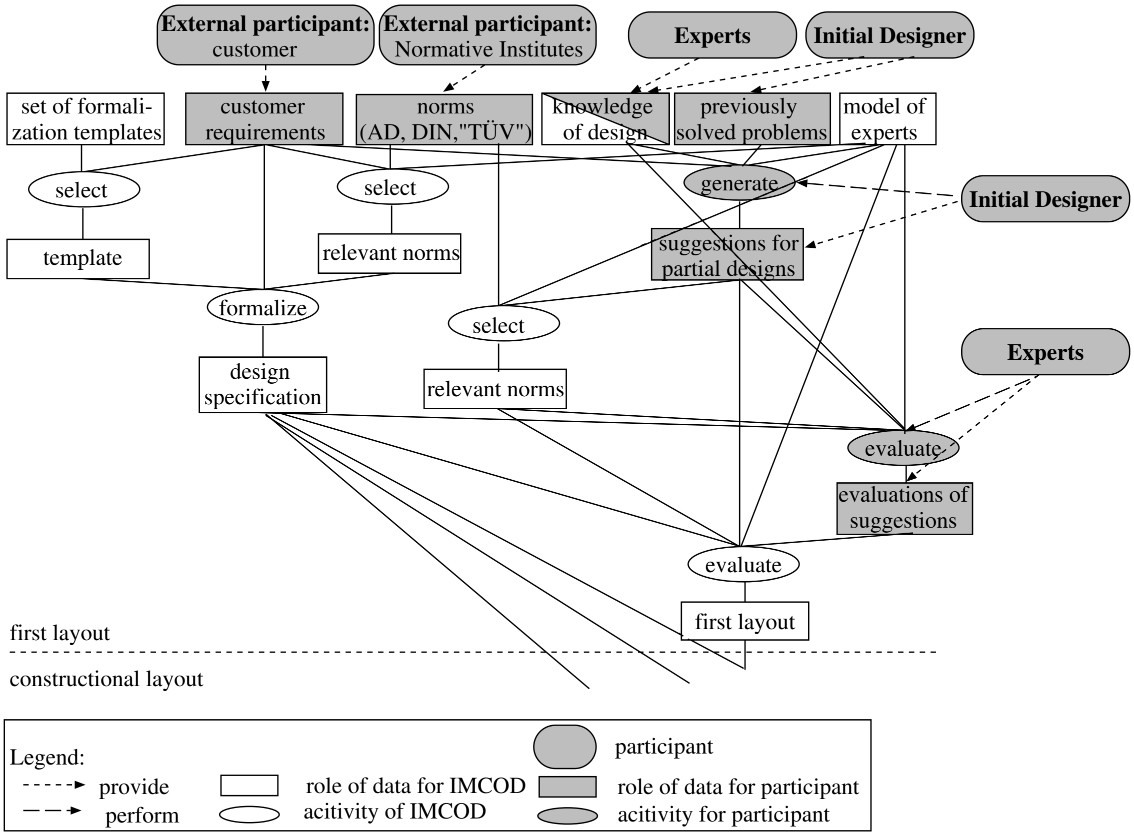

Figure 6: Enhanced inference structure of the first layout

Figure 6 shows the data needed/provided by IMCOD (white boxes) and by the other participants (grey boxes). The IMCOD system has to perform some activities of the design process (white ellipses) and the other activities are taken over by the other participants (grey ellipses). In order to preserve the clarity of the figure we refrain from illustrating which data is needed by a participant for solving a certain task. But, a participant needs all the (input) data elements which are connected with the respective activity. Furthermore,

[2]The KADS methodology offers no means to show this in the inference structure explicitly.

the special activity of IMCOD which describes the delegating and communication process is not explicitly shown, although it is always needed when one of the participants is involved in the problem solving process.
Figure 6 shows that the IMCOD system supports the development of a formal design specification from the informal customer requirements and norms. For the formalization the systems employs a set of formalization templates. Such templates are already known from the knowledge acquisition environment MIKADO–KIT [9]. The formalization step results in the formal design specification consisting of criteria and values which describe the customer's preferences. The creative task of finding suggestions for partial designs is delegated to the initial designer. As has been mentioned before, in our example this task must be solved by a human participant. The different experts are able to evaluate the design suggestions using the formal design specification. Based on the experts' evaluations the IMCOD system can select the best concept for the first layout.

5 The Construction Process for the Concrete Example

5.1 Formalizing the Requirements

The customer requirements are typically stated in informal text. To facilitate their evaluation by automatic means the requirements have to be transformed into some standardized and formalized representation. The result of this formalization step is the *design specification.*
The basic elements of the formalized design specification are the *design criteria.* These are extended attribute–value–pairs in the form

```
<identifier> <min-value>
             <max-value>
             <preferred-value>
             <tendency>
             <required?>
             <importance>
```

identifier is some text which unambiguously describes the attribute.

min–value, max–value define an interval in which shall contain the value of the attribute.

preferred–value describes the optimal value of the attribute.

tendency can be one of (-,0,+). This indicates the preferred deviation from the preferred-value.

required? indicates whether it is necessary for the design to fulfill the attribute. If this is set to NO, the attribute is considered *optional.*

importance only applies to optional criteria. It represents the importance of the attribute by a weight taken from a scale.

This representation of the design criteria facilitates the homogeneous handling of the different types of design criteria which are distinguished in general design methodologies [8].

5.1.1 Formalization templates

The formalization of the customer requirements into the design specification is supported by predefined templates of probable design specifications. A template consists of a set of design criteria which might contain values in any of the elements mentioned above.
The use of the formalization templates is twofold: The template guides the formalization of the customer requirements by providing typical criteria. The user can match the requirements to these criteria. If necessary the user may introduce additional criteria which are not contained in the template.
On the other hand the template serves as a representation formalism for basic technical knowledge. There exist certain criteria which the design has to fulfill but which aren't found neither in the customer requirements nor in the relevant norms, because they are considered basic technical knowledge. E.g. a pressure tank certainly must be impermeable to its filling, but this criterion is considered common sense and not stated in the requirements. The predefined template may contain a formalization of such criteria. Thus, the common sense knowledge is accessible to the automated design steps.
In our example, we defined two exemplary templates, one for cylindrical tanks, the other for spherical tanks. Both templates have most of the criteria in common, e.g. volume, pressure, recyclability .... The proposed geometry representation differs for both templates: The spherical tank only uses the primitves for ball joints to describe the complete form. Additionally, the cylindrical tank is described by primitives for cylinder surfaces and for arbitrary curvatures. Depending on the type of the tank, the AD guidelines provide different formulars for calculating forces and wall thickness.

5.1.2 The formalized requirements: The design specification

Using the template for cylindrical tanks the customer requirements are interactively formalized. This results in a set of design criteria, some of which are shown below:
Some Formalized Design Criteria for a Pressure Tank

```
'Pruefdruck': 17 bar (fix)
                 category: 'Dynamik'
                 required: true

'Lackhaftung': komplett (fix)
                 category: 'Gebrauch'
                 required: true

'Volumen': Interval [67 liter , 70 liter ]
                 category: 'Geometrie'
                 required: true
```

```
'Durchmesser': Interval [89 cm , 91 cm ]
                 category: 'Geometrie'
                 required: true

'Temperaturbestaendigkeit': Interval [-15 Grad , +40 Grad ]
                 category: 'Guete/Sicherheit'
                 required: true

'Abnahme TUEV': DIN (fix)
                 category: 'Guete/Sicherheit'
                 required: true

'Recyclierfaehigkeit': 80%(best) in: [40 % , 100 % ]
                 category: 'Stoffgroessen/chemisch'
                 defaultweight: 'very low'
                 required: false

'Grenzstueckkosten': 0 DM/Stck(best) in: [0 DM/Stck , 70 DM/Stck ]
                 category: 'Wirtschaft/Hersteller'
                 defaultweight: 'very low'
                 required: false

'Stossfestigkeit': 70 km/h(best) in: [50 km/h , 70 km/h ]
                 category: 'Stoffgroessen/ergonomisch'
                 defaultweight: 'very low'
                 required: false

'Wandstaerke': 1.1 cm(best) in: [1 cm , 1.4 ]
                 category: 'Geometrie'
                 defaultweight: 'medium'
                 required: false

'Produktionsanzahl': 8000(best) in: [7000 , 8000 ]
                 category: 'Fertigung'
                 defaultweight: 'high'
                 required: false
```

5.2 Initial Design Suggestions

The design of the pressure tank starts from some rough ideas about the general outline, the material to be used, and the production methods to be applied. As there are usually several possible alternatives to reach the intended product, it is important to allow a free flow of ideas at this stage of the design process. Existing design methodologies use a brainstorming phase, where every idea about the possible design is expressed and written

down. These ideas might be completely new and developed from scratch, or they might be adaptations of previous solutions.
The design suggestions are formally represented. Geometry, topology, materials, dimensions etc. are formalized into the product model. Additional information about the production methods etc. are represented by attribute–value pairs. Some values might contain arbitrary text, others will be accessible to later computation.
In our example, the brainstorming process resulted in six different design suggestions. Their main characteristics are:

1. Mandrel: sand-filled foil

   winding: fiber and polymer foil, then heat treatment

2. Prefabricated pipe + domes

   connection by glueing

3. Mandrel: solid from light material like Styrofoam

   winding: wet roving

4. Mandrel: metal, winding: wet roving

   Cutting the tank to remove the mandrel, then glueing

5. Liner made of thermoplastics

   Circumferential / helical winds with thermoplastic roving

6. Metal liner

   Circumferential / helical winds

5.3 Communication with the Participants

The interface between the manager — the IMCOD system — and the other participants is modeled with generic tools for two different layers: the *communication layer* [4] and the *semantics layer* [1]. Other translation tasks that have to be performed are not taken into account generically, e.g. the translation of different syntactical structures, since they have to be hard–coded in various algorithms. Therefore, we assume the existence of an interlingua in which the queries of the manager can be expressed and which can be syntactically understood by the other participants. However, since we assume the existence of the other participants which are not specifically tailored for our scenario, for each of them software modules must be realized that perform an additional syntactical translation task in the sense of the wrapper technology.

5.3.1 Modeling the ontologies

We assume that the ontology of the manger, i.e. the semantics of the identifier that are used while communicating with the other participants is already defined. A complete example of such an ontology can be found in [5]. The syntax and semantics for the representation

language in use is basically adopted from Ontolingua, which is supposed to be a de facto standard for representing ontologies.
As an example consider the following definition of a hierarchy of classes that is used in our scenario[3].

```
(define-class material (?mat)
  "A material has the properties
E-module and tensile-strength"
  :def (and (defined (material.e-module ?mat))
            (defined (material.tensile-strength ?mat))
            (defined (material.thermal-conductivity ?mat))
            (defined (material.thermal-expansion ?mat))
            ))

(define-function material.e-module (?mat) :-> ?emod
  "e-module is a vector-quantity of dimension 3 with
the pressure-dimension"
  :def (and (material ?mat)
            (e-module ?emod)  )  )

...

;;; Class Polymer
;;; subclass of Non-metal
(define-class polymer (?pol)
  ""
  :def (and (non-metal ?pol)))

;;; Class thermoplastics
;;; subclass of polymer
(define-class thermoplastics (?x)
    ""
  :def (and (polymer ?x))  )


...
```

These definitions basically form a hierarchy of concepts in which each concept describes the semantics of a kind of material as it is understood by the manager. Additionally, we assume that the *Material Expert* uses the lower part of the hierarchy with identical semantics. *Plastic* is supposed to be its root class. Hence none of these classes (more specifically: knowledge about the classes) need to be translated while the communication between the IMCOD manager and the other participant takes place.

[3]The example is adopted from [6]

The overall ontology is regarded as a first order theory that consists basically of

concepts (also: intensional types) *material*, *non-metal*, *plastic*, etc. which are regarded as unary predicates

constants *unrestricted*, *closed–circle*, *pk1060*

types (also: extensional types) *RecKind* defined as the set of its constants

relations the unary relation *recyclable* of the manager and the binary relation *recyclable* of the Recycling Expert

which forms the basic set of structuring elements for representing an ontology.
The ontology is sketched in figure 7.

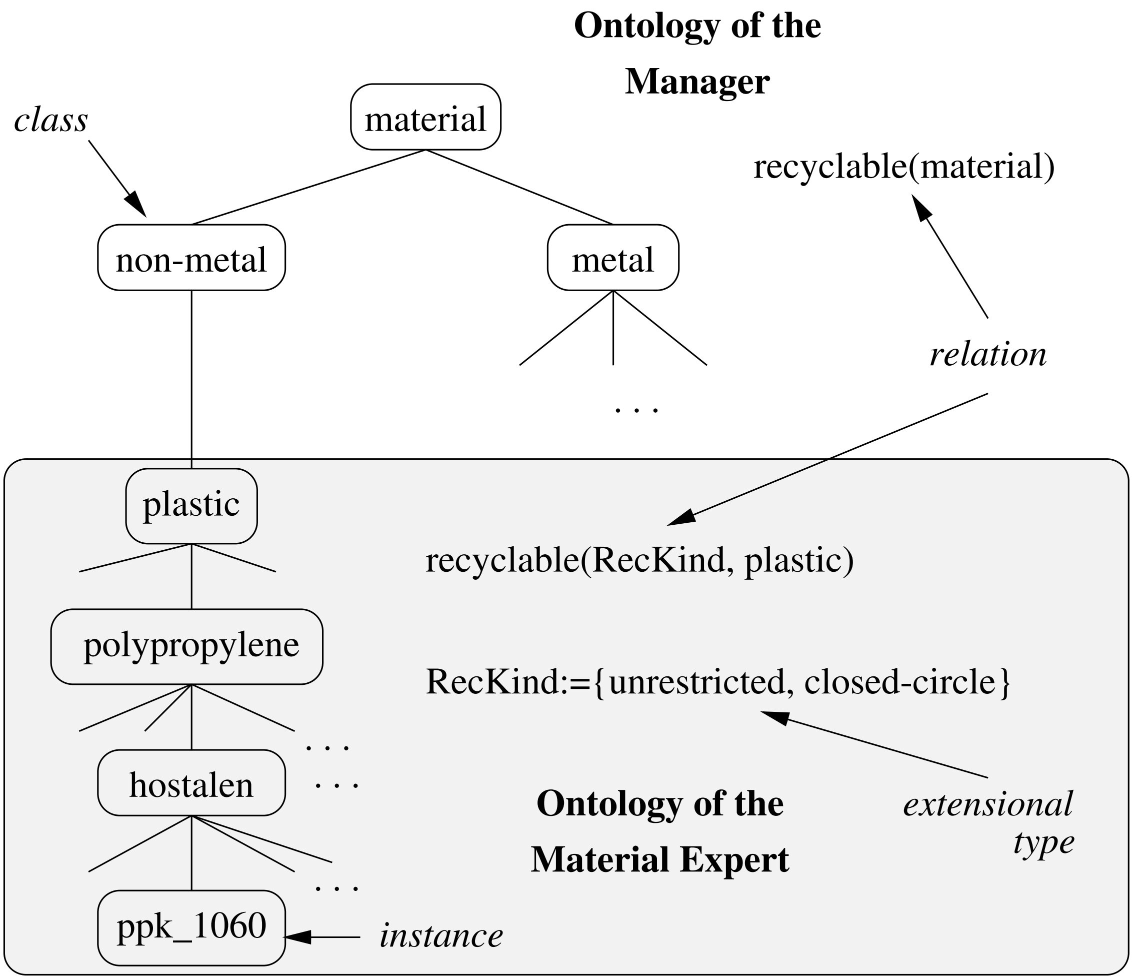

Figure 7: An ontology of materials (Manager and Material Expert)

5.3.2 Translation rules

Note that we use the relation *recyclable* for the manager and the Material Expert with different semantics. Therefore, we have to apply the method of semantic unification to enable the manager to ask the Material Expert such that the query has the same semantics for the local expert. For that purpose we apply the method of theory interpretation as an enabling technique for a constructive translation of formulas from one theory (in our terminology: ontology) to another. For a more complete description of this approach see [2].

A translation rule τ for a relation requires that we assign a formula to the predicate symbol in the other theory in which at most the variables of the relation occur free. In our example, a sensible translation rule for *recyclable* would be

$$\tau_{\text{recyclable}}(x) \quad := \quad \exists y.\text{recyclable}(x, y) \wedge (y = \text{unrestricted} \vee y = \text{closed–circle})$$

Here we see an example of a translation rule that intrinsically contains an abstraction (in the sense of Hobbs) which is a typical kind of knowledge that is expressed in a translation rule. Other kinds of knowledge are transformations, constant renaming, refinements, etc.

The ontology forms a hierarchy such that each identifier is only defined once in the complete hierarchy. In the above example we would start with the definition of the lower ontology that is equally used by the manager and the Material Expert and would define the relation *recyclable* individually in both ontologies—one for the manager and one for the Material Expert. Consequently, for all identifiers which occur in both ontologies the translation rule is a trivial one: the identity translation, e.g.

$$\tau_{\text{plastic}}(x) \quad = \quad \text{plastic}(x)$$

5.3.3 Modeling the interface

Additionally, we have to model the communication layer that defines how a message must be interpreted from a higher point of view: whether a new fact is told, whether the message must be interpreted as a query, etc. The messages at the interface are qualified via operators. We have considered five different types of operators. Besides the communication operators Tell and Ask we introduced three others: Retract, Command, Compose. Their semantics is explained in [4] and [3].

In the above example the manager can ask whether some kind of plastic can be recycled. With our set of of operators this question is classified as a *closed query* (a query yielding true or false as an answer) and is specified with our set of operators as follows:

Closed–Ask

language:	PROLOG
ontology:	PLASTIC–Ontology
documentation:	"Asks whether a specific kind of plastic is recyclable."
number-of-answer:	1
world-assumption:	CWA
destructive:	false
receiver:	Material Expert
content:	recyclable(?x:*plastic*)

Note that there are a number of parameters that give additional information of how the query and the corresponding answer have to be interpreted, i.e. that the answer has to be interpreted under the closed–world–assumption (CWA), that the knowledge base of the receiver is not modified by this query, etc.
In our scenario the manager holds a set of libraries, one for each of the other participants. Each contains a set of qualifiers. However, the content of the message is stated in the ontology of the manager and not in the one of the other participant.

5.3.4 Applying the model of the participant

So far we have only modeled the knowledge that is necessary to enable a semantically correct communication with the other participants. The question remains how such knowledge can be applied at run–time such that the manager poses a question to a participant, receives an answer and is able to interpret is correctly.
In our example, we assume that the manager wants to check whether an expert knows about a specific plastic and needs the information whether it (ppk_1060) is recyclable, i.e.

$$\exists x.\text{plastic}(x) \wedge x = \text{ppk\_1060} \wedge \text{recyclable}(x)$$

We describe the scenario as a sequence of steps that has to be performed:

1. The manager looks through the set of libraries to find out which expert is able to answer such a query. It will find the above entry whose contents contains the relation *recyclable* and which is qualified as a closed query.

2. The above translation rules (see 5.3.2) are applied to the content of the query such that we get a translated content that is semantically equivalent to the query but can be interpreted in the ontology of the receiving expert. The translation is structure preserving which means that formulas are to be applied according to the definition of FOPL.

   Since we assume that the unary relation *plastic* is used in both ontologies this relation is translated by the identity translation and we achieve the overall translated formula:

$$\begin{aligned}\exists x.x \;&= \text{ppk\_1060} \quad \wedge \text{plastic}(x) \\ &\qquad\qquad\qquad \wedge(\exists y.\text{recyclable}(x,y) \wedge (y = \text{unrestricted} \vee y = \text{closed–circle}))\end{aligned}$$

3. The following step would be a syntactical translation of the above expression in the interlingua into the specific syntactical structures of the other participant. Since this problem asks for very specific algorithms that typically contain hard–coded information this problem is not of much importance to us.

4. The translated example will then be sent by means of well defined technical interfaces, e.g. TCP/IP to the specific software module. Again since the problem here is not very challenging from an AI point of view and we are not looking for a generic approach. KQML (Knowledge Query and Manipulation Language) provides interfaces that can be used in this context.

5. The returned answer to this query (in our example: true) is then interpreted in the ontology of and by the manager, i.e. for the following design decision it assumes that the selected material holds the property and is recyclable. Note that the manager through the translation rules automatically abstracts from the additional information of the Material Expert whether the material can be recycled unrestrictedly or whether it can only be recycled in a closed circle due to problems of toxic dioxin on ultimate thermic treatment.

5.4 Evaluation Dialog

Using the communication interfaces IMCOD controls the evaluation dialog with the experts. The goal of this dialog is to let the different experts check whether the design suggestions fulfill the requirements of the design specification. Thereby, each expert is asked according to his special competence.
The competence of the experts is documented in the set of libraries which models the interface at the communication layer (cf. 5.3.3). From this information IMCOD knows how to communicate with an individual expert and which criteria can be evaluated by a particular expert. Thus, the design suggestions are distributed to the experts together with sets of criteria from the design specification. The experts state whether the suggestions fulfill the requirements, provide estimations about ranges of criteria (e.g. the probable cost of recycling), and calculate values.
The evaluation of an individual expert reflects the particular preferences of this expert. The amalgamation into a homogeneous assessment of a design suggestion requires two decision steps.

- For each criterion we distinguish two levels of competence of the relevant experts. The experts on the second level are only used if the first–level experts do not supply a satisfactory result.

- Afterwards, the assessment of a design suggestion with respect to all criteria is a multi–criteria decision task.

5.5 Multi–Criteria Decision

For each design suggestion the IMCOD system documents the experts' evaluation of the different criteria in a decision table. The table also contains the weights which indicate the

importance of the different criteria as they have been obtained during the formalization of the customer requirements.
The distinction between mandatory and optional criteria is of particular interest. The mandatory criteria represent hard constraints. If a mandatory criterion can not be fulfilled, this cancels the complete design suggestion. The optional criteria on the other hand are the objects of the optimalization by the multi–criteria–decision: The combination of the evaluations of the optional criteria with their respective weights provides the optimal design suggestion for the first layout.
To illustrate the decision process an exemplary decision table is shown in table 1. The table comprises three of the mandatory criteria and two optional criteria which are typical for our example.

Criteria	**Weight**	**DS 1**	**DS 2**	**DS 3**	**DS 4**	**DS 5**	**DS 6**
Tolerance	mandatory	0	1	1	1	1	1
Impermeability	mandatory	0	0	0	0	1	1
Security Test	mandatory	1	0	1	0	1	1
Production process	1.0 (high)	0.1	0.5	0.1	0.5	0.8	0.8
Costs of Recycling	0.3 (low)	0.8	0.6	0.6	0.3	0.8	0.4
$\sum$ **Assessments**		$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	1.04	0.92

Table 1: Multi–criteria decision for calculating the best concept for first layout

The design suggestions DS 1 — DS 4 are discarded because at least one of the mandatory criteria can not be fulfilled. For instance, the suggestion DS 1 (sand–filled foil as a mandrel) has been an interesting idea as it solves in an elegant way the problem of removing the mandrel from the final product. However a closer evaluation by the experts reveals that this approach does not allow satisfying tolerances. Likewise every suggestion which does not propose a liner as part of the final product proves to have insufficient impermeability. This result is not surprising in the application scenario: The brainstorming process was mainly concerned with rough and innovative ideas. The closer examination later on discards most of them at an early stage because they do not fulfill important criteria.
Among the remaining suggestions the multi–criteria decision provides the optimal one. In our example, we use a simple weighted sum to calculate the final assessment a :

$$a = \sum_{i=1}^{n} w_i * e_i$$

w_i is the weight of criterion c_i, e_i is the respective evaluation.

5.6 The Result: The Selected Design

The result of the decision process is the identification of design suggestion DS 5 — thermoplastic liner and thermoplastic roving — as the optimal one: The mandatory criteria are fulfilled (the liner provides both impermeability and acceptable tolerances), the costs of recycling are estimated better (because of the homogeneous material) than in the suggestion DS 6 (metal liner).

Thus, suggestion DS 5 is presented as the result of the first layout phase and can be calculated in detail during the constructional layout.

6 Discussion and Conclusion

In this paper we demonstrated the functionality of the IMCOD system using the example of designing a pressure tank made of composite materials. The goal of IMCOD is the support of a responsible product manager during the construction process. The main tasks tackled by IMCOD cover the control of the process, the communication to the participants in the scenario and the support of the homogeneous evaluation of the respective contributions.

During the process performed in the example several interesting scientific problems arised: The transformation of informal data, e.g. the customer requirements, the norms and technical guidelines, into a formal representation, i.e. the design specification, is a general problem of knowledge acquisition. This problem has been investigated in more detail during the development of the knowledge acquisition environment MIKADO-KIT [9].

The problem solving process which must be performed under guidance of the IMCOD system (the manager) can usually be described using a KADS model. However, the existence of several individual participants and the task to delegate jobs to these participants cannot be represented in the same KADS model. This tends towards an enhancement of the existing KADS libraries and merits further research.

The communication task leads immediately to the problem of semantic unification. The approach presented here is based on well–known interchange formats (e.g. Ontolingua) and allows the flexible representation of the ontologies and interface information of the individual participants. Using this information sets of translation rules can be defined. Together these components realize the theory interpretation necessary for the semantic unification.

The multi–criteria decision used in the example applies well–known techniques but requires their integration into a new context.

The abstract scenario of IMCOD, the distinction between a central manager and different participants contributing to the solving of a common problem is of general interest. It can be applied to every situation where multiple experts work on the solution of complex problems under the monitoring and guidance of one responsible entity. This especially covers many design problems.

The central competence of the IMCOD system lies on the managing aspect of the scenario. Thus, the system is not in competition with a CAD system or similar design support systems. However, such systems might be among the participants in the scenario.

Some of the concepts and methods presented here have already been implemented, the others are currently under implementation in a prototypical system.

References

[1] B. Bachmann. RanDom — Representation Language for Domain Theories. In K. Hinkelmann and A. Laux, editors, *DFKI–Workshop: Wissensrepäsentations–Techniken*, DFKI Document, July 1993.

[2] B. Bachmann. Mappings between ontologies in multidisciplinary design. In M. Alberts, editor, *Workshop Notes (AI & Design '94): A Semantic Basis for Knowledge and Data in Design*, 1994.

[3] B. Bachmann. Enabling knowledge exchange in multidisciplinary design support systems. *International Journal of Industrial Engineering*, 1995. submitted.

[4] B. Bachmann and F. Dridi. Definition of a communication layer for expert systems. In J. Chen, editor, *Proc. 6th Int. Conf. on AI and Expert System Applications, Houston, Texas*, 1994.

[5] B. Bachmann and M. Steffens. An ontology of materials. In *Workshop Notes (ECAI '94): Comparison of Implemented Ontologies*, 1994.

[6] K. Hinkelmann, M. Meyer, and F. Schmalhofer. Knowledge–base evolution for product and production planning. *AI Communications*, 7(2):98–113, 1994.

[7] F. Maurer. *Hypermediabasiertes Knowledge Engineering für verteilte wissensbasierte Systeme.* Number Band 48 in DISKI. infix, Ringstr. 32, D–53757 Sankt Augustin, 1993.

[8] K. Roth. *Konstruieren mit Konstruktionskatalogen.* Springer-Verlag, Berlin Heidelberg New York, 1982.

[9] G. Schmidt. *Modellbasierte, interaktive Wissensakquisition und Dokumentation von Domänenwissen (MIKADO).* submitted PhD thesis, Universität Kaiserslautern, Fachbereich Informatik, 1995.

[10] M. Steffens. Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff. Document D–94–02, DFKI GmbH, Kaiserslautern, Postfach 2080, D–67608 Kaiserslautern, 1994.

[11] B. J. Wielinga, A. T. Schreiber, and J. A. Breuker. KADS: A modelling approach to knowledge engineering. *Knowledge Acquisition*, 4(1), 1992.