Model-Based Chemical Compound Formulation

Stefania Bandini, Alessandro Mosca and Matteo Palmonari

Department of Computer Science, Systems and Communication (DISCo),
University of Milano-Bicocca, Milan, Italy
{stefania.bandini,alessandro.mosca,matteo.palmonari}@disco.unimib.it

Summary. Many connections have been established in recent years between Chemistry and Computer Science, and very accurate systems, based on mathematical and physical models, have been suggested for the analysis of chemical substances. However, such a systems suffer from the difficulties of processing large amount of data, and their computational cost grows largely with the chemical and physical complexity of the investigated chemical substances. This prevent such kind of systems from their practical use in many applicative domain, where complex chemical compound are involved. In this paper we proposed a formal model, based on qualitative chemical knowledge, whose aim is to overcome such computational difficulties. The model is aimed at integrating ontological and causal knowledge about chemical compounds and compound transformations. The model allowed the design and the implementation of a system, that is based on the well known Heuristic Search paradigm, devoted to the automatically resolution of chemical formulation problems in the industrial domain of rubber compounds.

1 Introduction

In recent years, many connections between Chemistry and Computer Science have been established in the context of several research areas (e.g. computational representation of atoms and molecules, the storing and searching for data on chemical entities, identification of the relationships between chemical structures and observable behaviors, theoretical elucidation of structures based on the simulation of forces). Researchers in the area of Computational Chemistry sought to develop theoretical and computational methods based on mathematical models for describing and understanding the movement and the function of electrons in molecules, and applied these methods to significant problems of broad chemical interest [1]. Indeed, the term "computational chemistry" is used when a *mathematical* method is sufficiently well developed that it can be automated for implementation on a computer [2–8].

Although such mathematical methods are well-known and there are a number of systems based on them, their computational cost grows largely with the

S. Bandini et al.: *Model-Based Chemical Compound Formulation*, Studies in Computational Intelligence (SCI) **64**, 413–430 (2007)
www.springerlink.com

number of electrons [9] and, therefore, with the chemical and physical complexity of the investigated chemical substance. Reasoning on the structural and behavioral change dynamics of chemical compounds (i.e. of chemical substances formed from two or more elements, with a fixed ratio determining the composition) is a hard combinatorial problem, even when a small number of chemical elements are taken into account.

A crucial problem in applied chemistry, in which the physical and chemical complexity of the involved substances can be extremely high is the chemical *Compounding Problem*. The chemical compounding problem consists in the task of generating in an automatic way new complex compound formulations on the basis of a set of desired final behaviors in order to support industrial production processes. The computational limitations of the actual computational chemistry systems suggest that we must rely on different modeling techniques. In other words, as far as the problem of designing and implementing systems that reason and drive transformations of complex chemical substances is concerned, it is a challenging task to overcome the computational intractability of the *quantitative*, mathematical, compound representations.

Now, two questions arise: (1) what does it mean to reason upon chemical compounds taking advantage of a formal model representing *non*-quantitative chemical knowledge (e.g. ontological and causal expert knowledge)? (2) What is the kind of formal representation that allows to automatically transform the formulation of compounds with respect to specific engineering objectives, still preserving their identity as particular chemical compounds (e.g. drug compounds or tyre rubber compounds)?

In the effort to answer these questions, our research led to an epistemological investigation of the qualitative knowledge characterizing chemical compounding problems, and to the definition of a formal representation of that knowledge. A definition of the compound formulation problem (or, compounding problem), stretching some characteristics that have a direct impact on its computational tractability, is given in the next section. The section contains also a brief review of two research areas that are strongly related to the present work. Section 3 provides an introduction to the kinds of knowledge that are involved into the chemical formulation activity, with a focus on the formal ontological axioms defining integrity conditions for the chemical rubber compounds. Section 4 concerns the representation of causal knowledge together with its integration with ontological knowledge into the state space. Concluding remarks end the paper.

2 A Computational Perspective on the Compounding Problem

In industrial domains, the compounding problem, whether for agrochemicals, pharmaceutical, or speciality chemicals areas, deals with the possibility of modifying the formulation of some existing *chemical compound*, in order to

gain new compound formulations showing final desired performances. New desired performances for a compound can be originated by specific marketing commitments, design and cost requirements, or by constraints induced by the production process. Desired final performances are always expressed in terms of performance variations with respect to the preexistent compound (e.g. the increase of the *Rolling Resistance* of a tire, together with a reduction of the *Wet Handling*, and the maintenance of all the remaining performances).

Performances are observable behaviors of the chemical compound that are evaluated by means of a specific set of laboratory tests (in which compound behaviors are evaluated in isolation) and environmental tests (in which compound behaviors are evaluated into the final using environment). The system should find a compound structure, whose associated behaviors fit the requirements.

Compounding problem never coincides with a *ex novo* generation of compounds: a problem begins with some form of product specification, together with the specification of new desired performances, and ends with one or more new product specifications that meet the requirements. Therefore, the problem concerns the discovery of a suitable set of transformations that can be applied to the compound formulation in order to obtain a new compound with a specific set of desired performances.

The hard combinatorial nature of the problem essentially depends on the complexity caused by the effects of the application of transformations in compound structures. According to an *holistic* perspective, a transformation in quality/quantity of the elements of a chemical compound (no matter how massive or tiny it is) implies a non-uniform rearrangement of *all* the values of its associated properties, and this makes really problematic to find good a sequence of transformations pointing to the final desired compound. For example, it is usual that the effects of a structural transformation return a compound performing only some of the requirements, and failing on the others; obviously, this situation needs to find further transformations to bridge the gap among the modified object and the desired goal, but there is no guarantee that such transformations exist. This characteristic is common to a number of formulation, design, configuration, or planning problems dealing with entities (e.g. chemical mixtures, blends, and compounds, industrial plant, car engines, rescue or process plans) whose inner structures can be articulated in *parts*.

Generally speaking, transformations on parts produce not uniform transformations of all the wholes' properties, and this makes the search of a sequence of transformations pointing to the solution really problematic (for example, it is usual that the effects of a transformation return a compound performing only some of the requirements, and failing on the others; this situation needs to find further transformations to bring the gap among the modified object and the desired goal, but there is no guarantee that such transformations exist). It is therefore a challenging objective from an AI computational perspective to discover, represent, and exploit domain-specific knowledge with the aim of reducing such an explicit combinatorial complexity.

Since the application of a structural transformation must be evaluated with respect to all the behaviors associated to the compound, a solution to a compounding problem is necessarily a *compromise solution*. In other words, in compounding the *optimum* does not exist. Specific ranges of tolerance have to be defined together with compounding requirements in order to increase the possibilities of bringing a solution. Therefore, the existence of a solution to an instance of the compounding problem is guaranteed only by the accuracy of the compounding expert requirements: if we are looking for an "extravagant" set of performances for chemical compounds devoted to a specific marketplace, there are no guarantees that the compounding will succeed.

2.1 Related Research Areas: Configuration and Planning

There are at least two well known research areas in Artificial Intelligence that are strongly related to the proposed definition of the compounding problem: the area of *Automated Configuration* and that of *Automated Planning*. Configuration and Planning are very close to Compounding, although they have characteristics that does not always perfectly match with our problem.

Configuration can be defined as the design of an individual product by using a set of pre-defined components or component types. Configuration takes into account a set of well-defined restrictions on how the components can be combined together [10]. Planning was emerged as a specific sub-field of Artificial Intelligence with the seminal work of Fikes and Nilsson [11] on the Stanford Research Institute Problem Solver (STRIPS). Newell and Simon's work on GPS [12], Green's QA3 [13, 14] and McCarthy's situation calculus [15] helped to define the classic planning problem and many of their assumptions still influence planning research today. Very briefly, a planning problem is described by a collection of actions, each characterized by their *pre*-conditions (what must be true in order for the action to be executed) and their *post*-conditions (which describe the effect of execution of the action), an initial state of the world and a description of the goals to be achieved. The problem is solved by finding actions that will transform the given initial state into a state satisfying the given goals

Traditional researches in automated configuration and planning have relied on simple and relatively unstructured models of the problem and have placed the emphasis on the development of more efficient algorithms and powerful heuristic control methods. Nevertheless, [16, 17] and others, have recognized that a model typically contains hidden structure that can be exploited by a planner and, under this assumption, several research communities have focused on exploring more articulated modeling choices with the extent of expediting the solution search (see, for example, [18, 19]). Closely related to this perspective are: (i) the logical approach suggested by Kauz and Selman, based on the notion of *Satisfiability* for propositional formulas [20]; (ii) the Lifschitz's approach [21–23], grounded on the *Answer Set Program-*

ming paradigm and related to the researches on *Stable Models* of Subrahmanian and Zaniolo [24]; (iii) the *Model-Checking* approach of Giunchiglia and Traverso [25, 26]; and (iv) the work of Eiter, Faber, Leone, Pfeifer, and Polleres on the $\texttt{DLV}^{\texttt{k}}$ system [27].

It is beyond the scope of this article to furnish an extensive analysis of the relationships between the compounding problem, on one hand, and configuration and planning problems, on the other (further details can be found in [28]). Nevertheless, we are convinced that most of researches on configuration and planning can be grouped together under the following statements, that actually represent also the background of our work: (i) *the more human knowledge and expertise are embedded in the domain* model*, the less discovery has to be made by the planner*, and (ii) *the correctness of the reasoning system fundamentally depends on the correctness of the* model.

3 Compounding Knowledge

Once a qualitative perspective has been assumed on the compounding problem, two main kinds of knowledge must be considered and integrated: *ontological* and *causal* knowledge. With ontological knowledge we refer to the knowledge that specifies what entities have to be considered as admissible compounds' structures and behaviors (establishing their "integrity conditions" with respect to a domain of interest). This knowledge concerns entities within different perspectives (structural and behavioral), and it guarantees that transformations applied to those entities preserve their ontological integrity. With causal knowledge we refer to knowledge mapping compound transformations at the structural level (i.e. on the compound formulation) to transformations at the behavioral one (i.e. on the tested performances). Causal knowledge in compounding allows to expect the changes on compound behaviors, on the basis of transformations of its chemical formulation.

As for the chemical engineering domain, the automated discovery of ontological and causal knowledge is a problem too hard due to the computational complexity issues (see Section 1). Nevertheless, this knowledge already lives (expressed in *qualitative* terms) in the expert compounding practices and communities, and it can be elicited and formally represented by means of knowledge engineering techniques. Expert knowledge on compounding is often *not* immediately quantifiable, *not* directly math-based, and *not* microscopic; this knowledge has been worked out in chemical industrial context during all the Twentieth Century, producing a number of results that have lead to the success of several Chemical Engineering applications [29, 30].

In this paper, we propose a knowledge model to tackle a compounding problem that is formally based on: (i) a *description logic* (DL) knowledge base (in the language $\mathcal{SHOIN}$), describing ontological representations of compounds' structures and behaviors; (ii) a causal knowledge formal representation, coded into *morphisms* that map structural and behavioral

compound representations, at one hand, and structural and behavioral transformations, at the other.

3.1 Mereological Axioms for Compounding

A rubber compound is usually viewed by chemical engineers as a "recipe" or as *a blend of atomic components composed in various proportions*. Atomic components are aggregated into several "systems", in accordance with the *functional role* they have to perform within the compound. Therefore, along this perspective, chemical compounds can be observed as "aggregate objects", a notion for which a wide philosophical literature and different mereological investigations exist (e.g. see [31–35]).

Compounding problems in industry are characterized by the presence of many different formulations for a compound composition; nevertheless, according to the final use of a compound, it is possible to identify a set of necessary boundaries within which all the admissible compound formulations must rely. These boundaries have been represented by means of a formal theory, written in logical terms. Logically speaking, the models of this theory are all those compound formulations that do not cross the chosen ontological boundaries and consequently do not violate ontological integrity constraints.

The model of the ontological knowledge we propose is grounded on a "composed" part-of relation $\prec$, in the sense of Sattler's taxonomy [36] (i.e. a part-of relation that is both *integral* and *functional*). In particular, $\prec$ is a *finite*, *irreflexive*, *asymmetric*, and *intransitive* binary relation. The assumption on functionality (and therefore, on the intransitivity of the part-of relation) is justified by the specific domain we are interested in: all the chemical entities into a compound play a specific functional role with respect to its constitution. DLs are a logic-based formalisms for the representation and reasoning about conceptual knowledge.

In DL, *concepts* are used to describe classes of individuals sharing common properties, and *roles* are used to represent binary relations.

Therefore, let $\prec$ be a primitive role of a DL language standing for "is a functional part of"; it is also useful to introduce the inverse role "has a functional part of" (or $\succ$) as $\succ \doteq \prec^{-1}$. Since functional *part-of* relation is "integral", for an entity to be part of another simply means that the entity must satisfy integrity conditions associated to the relation. Functional part-of is constrained to hold only among entities of a certain predefined kind. Here, the integrity conditions are simply expressed by means of different concept names and value restrictions of the form $\forall R.C$.

As far as compounding is concerned, we know that whatever is the compound of interest, its direct parts must be of type `System`, and that whatever is the system, its direct parts must be of type `GroundElement`. This means to assume in our ontological knowledge representation the following General Concept Inclusion (CGI) axioms:

$$(I1) \qquad \texttt{System} \sqsubseteq \forall \prec .\texttt{Compound}$$

$$(I2) \qquad \texttt{GroundElement} \sqsubseteq \forall \prec .\texttt{System}$$

Intuitively, the axioms say that a system may be only a part of a compound and a ground element may be only a part of a system. A so rigid hierarchy is coherent with the abstract representation of compounds as in real problem solving contexts (e.g. [30, 37]).

Ground Elements. Ground elements are the "atomic" entities living in the compounding domain (i.e. they have no parts). It seems reasonable to think that ground elements represent "minimal manageable quantities" of chemical substances: each of them represent a *fixed quantity* of a given substance, characterized by different chemical and physical properties. In concrete domains, ground elements are obviously chosen in accordance with chemical and physical properties.

Attributes. Attributes and properties of a ground element may be represented by introducing in the language specific DL roles, named *functional roles*. A role $\texttt{R}$ is said to be a functional role if and only if $\{(a,b),(a,c)\} \subseteq \texttt{R}$ implies $b = c$. Each concept is characterized by a suitable set of those functional roles. We indicated with `NumericalValue` a generic filler for functional roles (instances of this generic concept may be integer or real numbers, in accordance with the employed physical measurements). One can formally represent attributes of ground elements by instantiating the following axiom schema:

$$\texttt{GroundElement} \sqsubseteq f_1.\texttt{NumericalValue} \sqcap \cdots \sqcap f_n.\texttt{NumericalValue}$$

where $f_1, \ldots, f_n$ are n generic functional roles.

Exclusive Parts. Close to the axioms (I1) and (I2), it is useful to represent also exclusive relationships among these concepts. In general, a part is said to be "exclusive" if and only if there exists *at most* one whole containing it. Such feature expresses a kind of interdependence among whole and part. In compounding, the introduction of expressions about exclusivity of parts, forces models in having ground elements of certain kind (e.g. `NaturalRubber`, `CarbonBlack`) only as constituent parts of specific systems. Exclusive parts are formally represented in DL as particular instances of number restrictions. Number restrictions are concepts of the form $(\geq nr.C)$ (at-least restriction) or $(\leq nr.C)$ (at-most restrictions), where n is a non-negative integer, r is a role, C is a concept. In order to represent exclusive parts of a whole, it is possible to specialize number restrictions by means of the equality symbol $=$, stating that each `GroundElement` (`System`) is part-of exactly one `System` (`Compound`), as follows:

$$\texttt{GroundElement} \sqsubseteq (= 1 \prec .\texttt{System})$$

$$\texttt{System} \sqsubseteq (= 1 \prec .\texttt{Compound})$$

Upper and Lower Bounds. Systems may contain ground elements with different quantities. Ontological compounding knowledge provides *upper and lower bounds* of quantity of a ground element a system may contain. To represent the admissible range of quantity of an element is necessary to preserve the integrity of the compound during the formulation activity. The number restriction constructor allows to impose different range of quantities of an element: the $(\geq nr.C)$ and $(\leq nr.C)$ concept constructors can be combined in order to set upper and lower bounds, as follows:

$$\texttt{System} \sqsubseteq (\geq 1 \succ) \sqcap (\leq n \succ)$$

where n is an integer that will be instantiated according to concrete applications. Technically, the quantity of a substance in a system corresponds to the cardinality of the set of *has-part-of*-fillers of this system. From (I2) one can say that these fillers are all from the category of `GroundElement`. A system *must* have *at least* one ground element as its part, that is, a lower bound not inferior to 1. If we consider the relationships between `Compound` and `System`, the same situation arises: compounds may contain a number of systems, but *at least* one system must be contained. Therefore, ground elements must be considered as *essential parts* of systems, and systems as *essential parts* of compounds [38].

Atomicity. Atomicity immediately follows from the introduction of the ground elements. We resort to translate atomicity into the non-existence of fillers of the part-of relation in correspondence to ground elements. On the other hand, we can state that compounds cannot be part of any other entity.

$$\texttt{GroundElement} \sqsubseteq \forall \succ .\bot$$
$$\texttt{Compound} \sqsubseteq \forall \prec .\bot$$

In the rest of the paper, we present "tread tire compounds" as a paradigmatic example of chemical compound. The formulae we will introduce have to be understood as a specialization of the mereological theory introduced so far and, as a consequence, the involved concepts respect the ontological constraints.

3.2 The Case of Rubber Compounds for Tread Tire

The "tread" is the part of the tire in contact with the road. The profile and rubber compound are chosen on the basis of the use of the tire. The following logical formulae are a fragment of our mereology for compounding; in particular, the introduced formulae show some key elements of the ontological theory for tread rubber compound formulation (in what follows we grouped together entities that agree on the same mereological level). The set of axioms guarantees that if a model of these statements exists, then this model describes a compound for the production of tread tire in the industrial field of interest.

$$\begin{aligned}\texttt{TreadCompound} \equiv \texttt{Compound} &\sqcap (= 1 \succ .\texttt{PolymericMatrix})\\ &\sqcap (= 1 \succ .\texttt{Vulcanization})\\ &\sqcap (= 1 \succ .\texttt{ProcessAid}) \sqcap (= 1 \succ .\texttt{Antidegradant})\\ &\sqcap (= 1 \succ .\texttt{ReinforcingFiller})\\ &\sqcap ((\geq 0 \succ .\texttt{Softener}) \sqcap (\leq 1 \succ .\texttt{Softener}))\end{aligned}$$

It is standard that a rubber compound devoted to tread tire production is made of at least five essential systems [6, 37, 39, 40]: **(1)** the `PolymericMatrix` is the system that contains *polymers* (e.g. Natural rubber, Butadiene rubber, Styrenebutadiene rubber) and it plays a decisive functional role in tread compound. The final tread compound formulation will contain a suitable subset of discrete amounts of those polymers. **(2)** The `Vulcanization` system provides suitable chemicals for the compound vulcanization process. The system is made of "vulcanization chemicals" (e.g. `Sulphur`, `Peroxides`, `Urethane`), "vulcanization accelerators" (e.g. `Guanidines`, `Thiazoles`), "activators" (e.g. `MetalOxides`, `FattyAcids`, `SaltFattyAcids`), and "vulcanization inhibitors" (chemicals based on phthalimide sulfenamides). **(3)** The `ProcessAid` system, whose aim is to enable a rubber compound to be fabricated with less energy, is made of "peptizers" (e.g. `Renacit`) and "plasticizers" (e.g. `oil`); **(4)** the `Antidegradant` system is made of "antioxidants" and "antiozonats" that have been developed to inhibit the action of oxygen and ozone. Finally, **(5)** the `ReinforcingFiller` is defined as the ability of fillers to increase the stiffness of unvulcanized compounds, and the reinforcement effect of a filler shows up specially in its ability to change the viscosity of a compound; reinforcing fillers are `CarbonBlack` and `Silica`. Further systems may be present in a tread tire compound, such as the `Softener`, the `Extenders`, and the `Tackifier` systems, depending on the application context.

$$\begin{aligned}\texttt{PolymericMatrix} \equiv \texttt{System} &\sqcap (= 100 \succ .(\texttt{NaturalRubber} \sqcup \texttt{ButadieneRubber}))\\ \texttt{Vulcanization} \equiv \texttt{System} &\sqcap ((\geq 1 \succ .\texttt{Sulphur}) \sqcap (\leq n \succ .\texttt{Sulphur}))\\ &\sqcap ((\geq 1 \succ .\mathit{hasFamilyName}.\texttt{Accellerant})\\ &\sqcap (\leq m \prec .\mathit{hasFamilyName}.\texttt{Accellerant}))\\ &\sqcap ((\geq 2 \succ .\texttt{ZincOxide}) \sqcap (\leq p \succ .\texttt{ZincOxide}))\\ &\sqcap ((\geq 2 \succ .\texttt{StearicAcid}) \sqcap (\leq p \succ .\texttt{StearicAcid}))\\ \texttt{ProcessAid} \equiv \texttt{System} &\sqcap ((\geq 1 \succ .\mathit{hasFamilyName}.\texttt{Peptizer})\\ &\sqcap (\leq z \succ .\mathit{hasFamilyName}.\texttt{Peptizer}))\\ &\sqcap (= z \succ .\mathit{hasFamilyName}.\texttt{Plasticizer})\\ \texttt{ReinforcingFiller} \equiv \texttt{System} &\sqcap ((\geq 1 \succ .\texttt{CarbonBlack}) \sqcap (\leq n \succ .\texttt{CarbonBlack}))\\ &\sqcap ((\geq 1 \succ .\texttt{Silica}) \sqcap (\leq \frac{n}{2} \prec .\texttt{Silica}))\end{aligned}$$

The `PolymericMatrix`, the `Vulcanization`, the `ProcessAid`, and the `ReinforcingFiller` are systems. The polymeric matrix has 100 parts as a blend of natural and synthetic rubber or, alternatively, 100 parts of natural or synthetic rubber alone. Parts of the vulcanization system are the Sulphur, the Oxide Zinc and the Stearic Acid in a predefined *quantity*. The possibility of selecting parts by their membership to specific chemical families is

exploited in the definition of the vulcanization system. A vulcanization system contains some quantity of a (not further specified) element in the family of the `Accellerant`, while a process aid system takes part from the `Peptizer` and `Plasticizer` family. A reinforcing filler contains Carbon Black and Silica in a predefined *ratio*. The possibility of representing two quantities in a certain ratio is crucial in compounding: the presence of a ground element often asks for the presence of another one (e.g. the "activator-activated" couples of chemicals)[1].

$$
\begin{aligned}
\texttt{CarbonBlack} \equiv\ & \texttt{GroundElement} \sqcap (= 1 \prec .\texttt{ReinforcingFiller}) \\
& \sqcap\ hasSurfaceArea.\texttt{NumericalValue} \\
& \sqcap\ hasPorosity.\texttt{NumericalValue} \\
& \sqcap\ hasTortuosity.\texttt{NumericalValue} \\
\texttt{Renacit} \equiv\ & \texttt{GroundElement} \sqcap (= 1 \prec .\texttt{ProcessAid}) \\
& \sqcap\ hasDensity.\texttt{NumericalValue} \sqcap hasFamilyName.\texttt{Peptizer} \\
\texttt{NaturalRubber} \equiv\ & \texttt{GroundElement} \sqcap (= 1 \prec .\texttt{PolymericMatrix}) \sqcap hasStructure.\texttt{CIS} \\
& \sqcap\ hasMolecularWeight.\texttt{NumericalValue} \sqcap hasFamilyName.\texttt{Polymer} \\
\texttt{StyrenebutadieneRubber} \equiv\ & \texttt{GroundElement} \sqcap (= 1 \prec .\texttt{PolymericMatrix}) \\
& \sqcap\ hasMolecularWeight.\texttt{Value} \sqcap hasFamilyName.\texttt{Polymer} \\
\texttt{ButadieneRubber} \equiv\ & \texttt{GroundElement} \sqcap (= 1 \prec .\texttt{PolymericMatrix}) \\
& \sqcap\ hasStructure.\texttt{CIS} \sqcap hasFamilyName.\texttt{Polymer} \\
& \sqcap\ hasMolecularWeight.\texttt{Value}
\end{aligned}
$$

Carbon Black, *Renacit*, *Natural rubber*, *Butadiene rubber*, and *Styrene butadiene rubber* are ground elements and exclusive parts of the reinforcing filler system, the process aid system, and the polymeric matrix system, respectively. Carbon black is characterized by a specific value of "surface area", and by a specific "microstructure" (represented in term of its porosity and tortuosity). The Renacit is characterized by a specific value of "density", and by its membership to the family of Peptizers, while the Natural, Butadiene and the Styrene butadiene rubber by a "CIS" configuration, a specific molecular density and by their membership to the family of Polymers.

4 Causal and Ontological Knowledge into the State Space

As mentioned in the introduction, causal knowledge provides necessary information to compute (and, forecast) the application of compound structural transformations, on one side, and the effects these structural transformations have in behavioral terms (e.g. it is known that an increase of the amount

[1] Since the syntax of the $\mathcal{SHOIN}$ description logic does not allow to express individual variables, the m, n, p, z symbols need to be instantiated with appropriate integers once the axioms are taken to represent ontological knowledge in a specific compounding domain.

of Silica worsens abrasive and resistance behaviors of a tread rubber compound), on the other. Causal knowledge has been formally represented within the search paradigm [41–43], by means of a set of *transitions* of the state space, and a set of *morphisms* linking the different dimensions of which the space is made.

More precisely, the state space has been finally defined as the *product* of three different labeled transition systems, corresponding to three different levels of abstraction. The first one of these systems represents compound formulations: states are logical descriptions of concrete compound formulations, as introduced in Section 3.2, while transitions are transformations of these formulations (i.e. discrete *increases*, discrete *reductions*, and substitutions of ground elements). Transitions of this system can be formally represented as a set of functions from compound formulations to compound formulations, with (i) domain dependent *pre*-conditions, listing prerequisites that must be satisfied by the compound mereological structure, and (ii) *post*-conditions, which specify precisely how the structure must be transformed. For example, in what follows we sketch the definition of a transition representing an instance of the substitution class:

$$f^{\mathbf{NR}}_{cis_+}(r) \rightarrow r',$$

The function $f^{\mathbf{NR}}_{cis_+}$ returns a compound r', that is equal to r, except for the natural rubber of r, that has been substituted with an alternative natural rubber with a greater *cis* value (where "*cis*" refers to a basic property of polymers coming from the specific geometrical atoms arrangement). The satisfiability of the pre-conditions and the consistency of the application of the recipe transformations essentially depend on the mereological structure of the involved compound. In particular, a quantity increase of a *part* e cannot be applied to a given compound: (i) if some pre-conditions on its applicability are not satisfied (observe that these integrity constraints, rising from the semantics of the logical formulae introduced in Section 3.2, are imposed in order to discard the computation of usefulness compounds during the formulation activity), and (ii) if the effects of this application produce a new compound containing an amount of e that turn out to be outside its admitted range.

The second and the third labeled system represent the synthesis of two levels of behavioral evaluation of the compound. On one hand, a compound is evaluated by means of a specific set of mechanical laboratory tests that return *quantifiable* properties. On the other one, the final performances evaluation is provided by means of tests studying the interactions of the compound within its application environment and under different conditions (the final performances of a tread rubber compound, as an example, are evaluated under wet and dry road conditions, irregular terrains, extreme temperatures, and so on). The formulation expert knowledge has specific heuristics to trace back the *qualitative* results of these tests to the behaviors of a single compound or of an identified aggregate of compounds. Qualitative information about final performances of a compound can be thus inferred and computationally

managed, once a suitable metric has been provided with the help of expert chemical engineers.

The knowledge on how a chemical compound need to be modified in order to obtain a new compound with final desired properties essentially concerns the applicability of transformations at the structural level, and the effects they have on the associated behavioral levels. Once the three labeled transition systems have been defined, the problem solving knowledge can thus be understood and formally represented by means of a couple of *morphisms*, mapping states to states and transitions to transitions of the different systems.

A morphism $\Gamma \rightarrow \Gamma'$ between transition systems can be intuitively introduced as a pair (σ, λ), where σ is a function on states, preserving initial states, and λ is a partial function λ on the transition labels. The morphism maps a transition of Γ to a transition of Γ', whenever this makes sense; in other words, if (p, α, q) is a transition in Γ then $(\sigma(p), \lambda(\alpha), \sigma(q))$ is a transition in Γ' provided that $\lambda(\alpha)$ is defined.

The role we assign to morphisms here is strongly connected to the task of relating transformations at one representation level with transformations at the other ones. Morphisms carry expert causal knowledge linking structural transformations on the compound to behavioral ones and, therefore, they allow to forecast behavioral changes of a chemical compound during the searching activity of new formulation (e.g. morphisms represent the fact that an increase of the amount of Silica worsens abrasive and resistance behaviors, by appropriately mapping the structural transformation "Silica Increase" to "Abrasive Decrease" and "Resistance Decrease" behavioral transformations). Observe that the definition of these compounding morphisms depends exactly on the acquired expert problem-solving causal knowledge.

We omit here the formal definition of the product of labeled transition systems [28], but we furnish a diagrammatic representation of it in the figure below. Figure 1 shows the structure we obtained by connecting the three systems we mentioned above, and the resulting state space in which our compounding system operates.

The states $s = \langle c, l, h \rangle$ and $s' = \langle c', l', h' \rangle$, for which the three dimensions are represented in Figure 1, are elements of the state space (c stands for $c$ompound structure, l for $l$ow-level behaviors and h for $h$igh-level behaviors or compound performances). τ, τ' are compounding morphisms such that $\tau(s_c) = s_l$ and $\tau'(s_l) = s_h$; these morphisms represent that a specific compound formulation s_c is associated to compound behaviors s_l and compound performances s_h. Note that the association is plainly given at the beginning of the compounding problem, where a compound structure together with its associated behavioral evaluations is given as input of the problem. On the other hand, the association must be computed during the solution searching process, on the basis of the structural transformations the initial compound is subject step by step. In fact, the existence of morphisms representing causal knowledge in compounding is mandatory for the construction of new states and transitions in the structure of the state space.

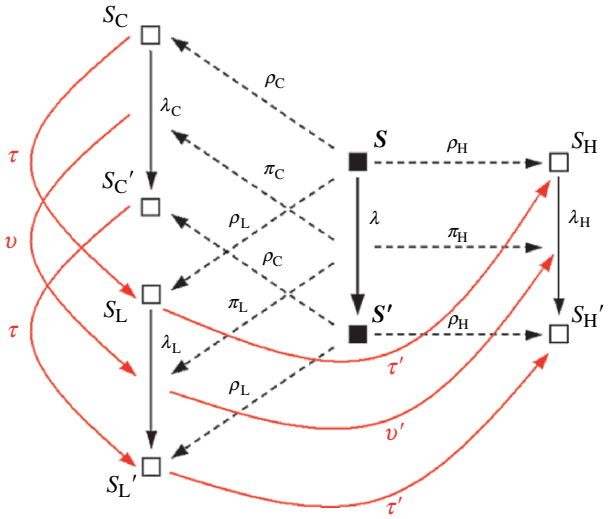

Fig. 1. Diagrammatic representation of the state space.

In Figure 1, an illustration of the compounding morphisms is given, with respect to projection morphisms of the product. The application of the transition λ to s, returning a new state s', is generated first of all by the application of a compound structural modification:

$$\lambda_C \colon S_C \to S_C$$

such that $\lambda_C(s_c) = s'_c$. The application of λ_C leads to a partial state $\langle c', l, h \rangle$, that it is not well defined, because it does not respect the constraints coming from causal knowledge and coded into the inter-dimensional morphisms. Therefore, in order to obtain a new correct state s', representing a feasible solution of the compounding problem, the transformations associated to λ_C have to be applied in the remanent behavioral dimensions. We can say that (s, λ_i, s') is a transition of the state space, written,

$$(s, \lambda, s') \in \rightarrow$$

if and only if $s' = \langle c', l', h' \rangle$, and $\tau(s'_c) = s'_l$ and $\tau'(s'_l) = s'_h$. Given by the introduced morphisms, the state components l' and h' are obtained by mapping the transition λ_C to transitions λ_L and λ_H.

5 A System for the Rubber Compound Formulation

Heuristic search algorithms occupy a fundamental place among all the artificial intelligence problem solving methods; these algorithms explore a solution space, in order to find optimal solutions to a given problem [42]. They require

a representation of (1) a *state space*, and (2) the choice of a *search algorithm*, possibly relying on good heuristic functions; moreover, their efficiency strongly depends on the involved formal representations.

The model we introduced combines ontological and causal knowledge in order to compute feasible solutions to the compounding problem. In this direction, a *compounding problem* is an instance of the state space introduced in Section 4, together with an initial state and a goal state. In the case of chemical compound formulation, the initial state would be any triple $\langle c, l, h \rangle$, provided that $\tau(c) = l$ and $\tau'(l) = h$. The *goal state* is partially specified in terms of required compound behaviors and performances: no information about the compound formulation performing these behaviors and performances is available as a component of the goal state.

From our computational perspective, if no ontological information on the states had been provided, every state in the state space would be generated and explored by the search algorithm as a feasible solution of the problem. On the contrary, with the support of the ontological representation, a state s' representing a compound formulation that resolves to be ontologically inconsistent (i.e. a formulation that is inconsistent with the ontological axioms of Section 3) is discarded by the system (i.e. pruned from the search tree). Since efficiency constraints do not allow to exploit an automated DL reasoner for checking the consistency of a compound formulation with respect to the axioms, compound formulations have been coded in an object-oriented data structure and the ontological constraints have been coded as pre- and post-conditions of the transition operators.

The proposed knowledge model, not only provides a sound representational framework for the state space, but it also allows to reduce the complexity of the solution space, exploiting the integration of ontological and causal knowledge. In fact, following both the constraints coming from the ontological representation of the chemical compounds, and the mapping between structural and behavioral transformations, the expansion of the search tree is exempted from computing useless ontologically inconsistent chemical compounds. In other words, all the possible expansions for the tree must respect the ontological consistency requirements of the involved compounds (e.g. if the given compounding problem concerns rubber compounds for tyre industry, a chewing gum must not be computed as "feasible solution" in the search tree). This reduces the combinatory of the searching process and minimize the computational effort of the implemented system.

The above computational model has been already exploited in solving the Chemical Formulation Problem in the domain of "rubber compound" production for tire industry [28] (this work has been part of the larger project "P-Truck", made in collaboration with the Business Unit Truck of Pirelli Tire S.p.A. [44]). In this context, a specific experimental campaign has been devised and encouraging experimental results have been obtained from the application of several search algorithms (namely A*, IDA*, Iterative Expansion and Branch and Bound) to a state space defined and implemented according to

the present knowledge model [45]. The IDA* algorithm has proved to be suitable for solving this kind of problem and, actually, more efficient and faster than the other experimented techniques. The results obtained by means of this algorithm have actually enabled the construction of new and performing rubber compounds. In particular, an automatic system has been developed and successfully tested on a significant number of prototypical chemical compounding problems (e.g. the problem of increasing the *Tread Tear Resistance* or, in a slightly more elaborate case, the problem of increasing the *Rolling Resistance*, together with a reduction of the *Wet Handling*, and the maintenance of all the remaining performances).

6 Concluding Remarks

Our research has been addressed to exploit a knowledge model in order to design and implement a system based on the Search paradigm. The system is devoted to perform searching in the chemical engineering area, improving its efficiency by suitable knowledge-based heuristics. This means that the integration between ontological and causal knowledge into our model produces immediate effects on the expansion rate of the search tree, with a considerable reduction in the time and space consumption for the system.

Recently, we are also engaged in investigating the use of Genetic Algorithms (GAs) [46, 47] to solve this kind of formulation problem. Compared to the other techniques that have been presented in this paper, GAs have the advantage that they enable to navigate even huge state spaces in an intelligent and efficient way, by means of a set of stochastic operators based on the Darwinian principles of biological evolution applied to a population of potential solutions. Compared to deterministic algorithms such as A*, IDA*, Iterative Expansion and Branch and Bound, GAs usually don't consider a large number of possible solutions, which are cut off the search process by means of a selection strategy which emulates natural selection. Several possibilities exist to apply GAs on the compounding problem, depending on which structures are chosen as the potential solutions to be evolved (or *individuals*, according to the GAs terminology). The use of Genetic Algorithms for the Compounding Problem is also motivated by the fact that classical AI algorithms generally work on decision tree structures and cut off the search process some subtrees, depending on some conditions. The eventuality that one of that subtrees contains one optimal solution is not remote, especially when large search spaces are considered. Working with a population of potential solutions, and being based on stochastic operators, GAs should enable an intelligent exploration of larger regions of the search space. In other words, the advantage of GAs for formulation problems using large quantities of data should not only be a lower computational effort, but also a higher quality of the solutions found.

References

1. Grant, G., Richards, W.: *Computational Chemistry*. Volume 29 of Oxford Chemistry Primers. Oxford University Press (1995)
2. Cohen-Tannoudji, C., Diu, B., Laloe, F.: *Quantum Mechanics* Volume I & II. John Wiley & Sons (1977)
3. MacQuarrie, D.: *Quantum Chemistry*. Prentice Hall (1983)
4. Hammond, B., Lester, W., Reynolds, P.: *Monte Carlo Methods in Ab Initio Quantum Chemistry*. World Scientific (1994)
5. Parr, R., Yang, W.: *Density Functional Theory of Atoms and Molecules*. Oxford University Press (1989)
6. Hoffmann, W.: *Rubber Technology Handbook*. Oxford University Press, New York (1989)
7. Burkert, U., Allinger, N.: *Molecular Mechanics*. American Chemical Society (1982)
8. Schlick, T.: *Molecular Modeling and Simulation*. Springer Verlag (2002)
9. Young, D.: *Computational Chemistry : A Practical Guide for Applying Techniques to Real World Problems*. Wiley-Interscience (2001)
10. Mittal, S., Frayman, F.: Towards a generic model of configuration tasks. In: *Proc. of the 11th IJCAI*, Detroit, MI (1989) 1395–1401
11. Fikes, R., Nilsson, N.J.: Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* **2** (1971) 189–208
12. Newell, A., Simon, H.A.: Gps, a program that simulates human thought. In Feigenbaum, E.A., Feldman, J., eds.: *Computers and Thought*. McGraw-Hill (1963) 279–293
13. Green, C.C.: Theorem proving by resolution as a basis for question-answering systems. In Meltzer, Michie, eds.: *Machine Intelligence* 4. Edinburgh University Press, Edinburgh (1969)
14. Green, C.C.: Application of theorem proving to problem solving. In: *IJCAI1*. (1969) 219–239
15. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., Michie, D., eds.: *Machine Intelligence* 4. Edinburgh University Press (1969) 463–502
16. Fox, M., Long, D.: The automatic inference of state invariants in tim. *Journal of AI Research* **9** (1998) 367–421
17. Gerevini, A., Schubert, L.: Inferring state constraints for domain-independent planning. In: *AAAI '98/IAAI '98: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial intelligence*, Menlo Park, CA, USA, American Association for Artificial Intelligence (1998) 905–912
18. Borrett, J.E., Tsang, E.P.K.: A context for constraint satisfaction problem formulation selection. *Constraints* **6** (2001) 299–327
19. Westfold, S., Smith, D.: *Synthesis of efficient constraint satisfaction programs* (1998)
20. Kautz, H., Selman, B.: Planning as satisfiability. In: ECAI '92: *Proceedings of the 10th European conference on Artificial intelligence*, New York, NY, USA, John Wiley & Sons, Inc. (1992) 359–363
21. Lifschitz, V.: Answer set programming and plan generation. *Artificial Intelligence* **138** (2002) 39–54

22. Lifschitz, V.: Answer set planning. In: *ICLP*. (1999) 23–37
23. Lifschitz, V., Turner, H.: Representing transition systems by logic programs. In: *LPNMR*. (1999) 92–106
24. Subrahmanian, V.S., Zaniolo, C.: Relating stable models and ai planning domains. In: *ICLP*. (1995) 233–247
25. Giunchiglia, F., Traverso, P.: Planning as model checking. In: *ECP '99: Proceedings of the 5th European Conference on Planning*, London, UK, Springer-Verlag (2000) 1–20
26. Spalazzi, L., Traverso, P.: A dynamic logic for acting, sensing, and planning. *Journal of Logic Computation* **10** (2000) 787–821
27. Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A.: A logic programming approach to knowledge-state planning, ii: the dlvk system. *Artificial Intelligence* **144** (2003) 157–211
28. Mosca, A.: A theoretical and computational inquiry into the Compounding Problem. Ph.D. thesis, Department of Computer Science, Systems, and Communication - University of Milano-Bicocca, Italy (2005)
29. Himmelblau, D.M., Riggs, J.B.: *Basic Principles and Calculations in Chemical Engineering*. 7 edn. Prentice Hall Professional Technical Reference (2003)
30. Duncan, T.M., Reimer, J.A.: *Chemical Engineering Design and Analysis, An introduction*. Cambridge Series in Chemical Engineering. Cambridge University Press (1998)
31. Fine, K.: Compounds and aggregates. *Nous* **28** (1992) 137–158
32. Husserl, E.: Logische Untersuchungen. Zweiter Band. Untersuchungen zur Phnomenologie und Theorie der Erkenntnis. Halle: Niemeyer (1900/1901) [2nd ed. 1913; Eng. trans. by J. N. Findlay: *Logical Investigations*, Volume Two, London: Routledge & Kegan Paul (1970)
33. Rescher, N.: Axioms for the part relation. *Philosophical Studies* **6** (1955) 8–11
34. Montague, R.: On the nature of certain philosophical entities. *The Monist* **53** (1969) 159–194
35. Simons, P., Dement, C.: Aspects of the mereology of artifacts. In Poli, R., Simons, P., eds.: *Computers and Thought*. Kluwer Academic Publishers (1996) 255–276
36. Sattler, U.: Description logics for the representation of aggregated objects. In W.Horn, ed.: *Proceedings of the 14th European Conference on Artificial Intelligence*, IOS Press, Amsterdam (2000)
37. Gent, A.E.: *Engineering with rubber, how to design rubber components*. Hanser Publisher, New York (1992)
38. Simons, P.: *Parts: A Study In Ontology*. Clarendon Press, Oxford (1987)
39. White, J.L.: *Rubber processing, Technology - Materials - Principles*. Hanser Publisher, Munich Vienna New York (1995)
40. Roberts, A.D., ed.: *Natural rubber science and technology*. Oxford University Press, Ney York (1988) s. 161.
41. Newell, A., Simon, H.A.: Computer science as empirical inquiry: symbols and search. *Commun. ACM* **19** (1976) 113–126
42. Korf, R.E.: Artificial intelligence search algorithms. In: *Algorithms and Theory of Computation Handbook*, CRC Press, 1999. CRC Press (1999)
43. Korf, R.E.: Search: A survey of recent results for Artificial Intelligence. In Shrobe, H.E., A.A., eds.: *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*, San Mateo, CA, Kaufmann (1988) 197–237

44. Bandini, S., Manzoni, S., Sartori, F.: Knowledge maintenance and sharing in the KM context: the case of P–Truck. In Cappelli, A., Turini, F., eds.: *AI*IA 2003: Advances in Artificial Intelligence, Proceedings of 8th Congress of the Italian Association for Artificial Intelligence*, Pisa (I), September 2003. Volume 2829 of Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Springer-Verlag (2003) 499–510
45. Bandini, S., Mosca, A., Vanneschi, L.: Towards the use of genetic algorithms for the chemical formulation problem. In Manzoni, S., Palmonari, M., Sartori, F., eds.: *Proceedings of the 9th Congress of the Italian Association for Artificial Intelligence (AI*IA* 2005), Workshop on Evolutionary Computation (GSICE 2005), Milano, Centro Copie Bicocca (2005) ISBN 88-900910-0-2.
46. Holland, J.H.: *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, Michigan (1975)
47. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley (1989)