# Using Concept Algebra for probing consistency and completeness of KAMET II, a Conceptual Modeling Language

**Osvaldo Cairó, Karen L. Poblete**

Department of Computer Science, ITAM, México DF, México

Address: Río Hondo 1, 01080 Mexico DF, Mexico

E-mail: cairo@itam.mx

## Abstract

KAMET II is a complete visual language with high usability and flexibility devised to acquire and organize knowledge from different sources in a very intuitive way. Similar recent languages used for knowledge representation are due to machine interpretation of knowledge without considering the users interaction. KAMET II, on the contrary, cares about the input facilities for constructing knowledge models without disregarding its compatibility with different methodologies for using and sharing the models. We describe and demonstrate the advantages of using KAMET II by probing it completeness and consistency by using Concept Algebra, a mathematical structure for the formal treatment of concepts and their algebraic relations, operations and associative rules. Concept Algebra provides a denotational mathematic aimed for knowledge system representation and manipulation. We do a direct transformation of KAMET II diagnosis models to Concept Network diagrams making use of Concept Algebra. As results, KAMET II models are compatible with regular Ontology representations and can be share and used by other systems.

**Keywords:** knowledge representation, knowledge ontology, knowledge conversion, knowledge modeling.

## 1. Introduction

Knowledge Based Systems (KBS) have been gaining a bigger market in different fields including commerce, science and government during the past few decades. Due to the high demand of KBS, different Methodologies for acquiring, validating, verifying and maintaining them had shown in the scene as useful tools to complete the Knowledge Engineering Process (KEP) of the KBS. The knowledge possessed by the human experts is often tacit and unstructured; the major goal of KEP is to help the experts to make explicit what they know and document the knowledge in an understandable and sharable manner.

KEP is divided in five major activities:

1. **Knowledge Acquisition (KA)**, which involves acquiring the knowledge from human experts, books, documents, sensors, or computer files. The knowledge can be specific to the problem domain or to the problem-solving procedures. Byrd

(1995) formally probed that KA is still an important problem in Expert System (ES) and KBS development today. A deep analysis of ES applications and their knowledge acquisition techniques and methods is available in Wagner et al. (2003).

2. **Knowledge Representation (KR)**. The acquired knowledge is organized to make it ready for use. This activity involves creating a knowledge map and encoding it in the knowledge base.
3. **Knowledge validation**. Involves validating and verifying the knowledge using commonly test cases.
4. **Inferencing**. Designing the software to compute inference chains to get results from a query given by a user. The solutions given are based on the stored knowledge.
5. **Explanation and justification**. Every answer given by the KBS can be explained, the system has a justification capability to support its answers.

KAMET II is a proposed methodology in (Cairó, Guardatti, 2012) due to assist the modeling of KBS during the knowledge acquisition and representation process of KEP. KAMET II life-cycle model provides a graphical framework for managing KA with structures to organize knowledge acquired from distinct sources, to share knowledge, monitoring project progress and to check quality controls. Here we focus on the knowledge representation process with KAMET II Concept Model Language (CML), a visual language for KR of diagnosis problems.

Knowledge modeling is a topic discussed by many authors, (Rincon et al., 2005), (Musen et al.,1999), where the use of an interactive knowledge editing tools allows the user to have a better experience while the KA is taking place. Recent work includes knowledge modeling on the Web (Davis et al., 2003) and Semiautomatic Knowledge Modeling Methods, where KA can be supported by computer-based tools, like an environment in which knowledge engineers or experts can identify knowledge through an interactive process.

KAMET II CML makes the process of KR easier by giving the user a visual tool, where the components of the models are just drag and drop to their places, it also represents a modern approach to set diagnosis-specialized models providing a user friendly framework.

Some recently languages used for knowledge representation are not intuitive; they are directed to computational interpretation than user interaction. The more recent methodologies for knowledge representation includes using Semantic Web and languages based in Extensible Markup Language (XML), including Resource Description Framework (RDF) (Lassila and Swick, 1999) Ontology Inference Layer (OIL) (Fensel et al., 2001), and Web Ontology Language (OWL) (Antoniou and van Harmelen, 2004). These languages that mainly focus on the creation of machine-readable representations for the Web, emphasize simplicity, generality, and usability over the Internet not taking care of the usability and the user level interactions. KAMET II CML is consistent and complete like the languages mentioned before, adding the characteristic of being an intuitive tool for knowledge modeling. KAMET II CML can be probe to be consistent and formal by using Concept Algebra (Wang, 2008). We use Concept Algebra to translate KAMET II CML models to an abstract algebra representation of the knowledge: concept networks, which represent a set of concepts and their relations.

Concept Algebra provides generic and formal knowledge manipulation tools capable to deal with complex knowledge and software structures as well as their algebraic operations. It can be applied to formalize object-oriented methodologies and to develop rigorous semantics of UML as an industrial OO design language. Knowledge can be manipulated by using concept algebra to build concept networks which can be analog to KAMET II CML models. A concept is a cognitive unit to identify and/or model real-world concrete entity and a perceived-world abstract subject (Wang, 2008), KAMET II CML nodes are analog to concepts and objects and we are going to show it later on. Each component of KAMET II models can be translated in an appropriate object, concept or attribute, also de attribute of accuracy has its representation, so do the cardinality closures that represent where a certain quantity of symptoms, problems or blocks have to be present to infer if a problem is happening .

In section 2 we introduce the basis of Concept Algebra needed to demonstrate the congruency of KAMET II CML and its assumptions. In section 3 we introduce KAMET II CML, its components, structure and rules of construction. Over section 4 we proceed to demonstrate the KAMET II CML consistency and to show some examples. We leave future work and conclusions section 5 and 6.

## 2. Concept Algebra for managing Knowledge Representation

In order to demonstrate KAMET II consistency and transparency we made use of Concept Algebra which is an abstract mathematical structure for the formal treatment of concepts and their algebraic relations, operations and associative rules to composing complex concepts. Concept Algebra was developed by Yingxu Wang and provides a denotational mathematic means for knowledge system representation and manipulation. KR is centered by model of concepts, which are the mean units of thought and reasoning.

A *concept* is a cognitive unit to identify and/or model a real-world concrete entity and perceived-world abstract subject. It is an abstract structure of tangible entities or intangible ideas, concepts can be classified into two categories, concrete or abstracts. A concrete concept represents an entity from reality while and abstract concept are for intermediate representation in reasoning.

Concept Algebra consists of three parts known as the mathematical model of abstract concepts, relational operations, and compositional operations. An abstract concept $c$ is a prime model of concept algebra, which is defined as a 5-tuple, i.e. $c \triangleq (O, A, R^c, R^i, R^o)$(Wang, 2009). The relational operations define operations that do not modify the concept but point the dependencies and comparison between concepts, subconcepts and superconcepts. The compositional operations provide a set of manipulation of abstract concepts like inheritance, composition and aggregation. For more information refer to (Wang, 2008).

### *3.1. The Abstract Model of Concepts*

In this section we make an abstract of the bases of de Concept Algebra model.

**Definition 1.** Let $\mathcal{O}$ denote a finite nonempty set of *objects*, and $\mathcal{A}$ be a finite nonempty set of *attributes*, then a *semantic environment* or *context* $\Theta_c$ is denoted as a triple, i.e.:

$$\begin{aligned} \Theta_c &\triangleq (\mathcal{O}, \mathcal{A}, \mathcal{R}) \qquad (1) \\ &= \mathcal{R}: \mathcal{O} \to \mathcal{O} | \mathcal{O} \to \mathcal{A} | \mathcal{A} \to \mathcal{O} | \mathcal{A} \to \mathcal{A} \end{aligned}$$

where $\mathcal{R}$ is a set of relations between $\mathcal{O}$ and $\mathcal{A}$, and | demotes alternative relations.

**Definition 2.** An *abstract concept c* on $\Theta_c$ is a 5-tuple, i.e.:

$$c \triangleq \left(O, A, R^c, R^i, R^o\right) \qquad (2)$$

where

- $O$ is a finite nonempty set of objects of the concept, $O = \{o_1, o_2, \dots, o_m\} \subseteq \text{Þ}\mathcal{O}$, where $\text{Þ}\mathcal{O}$ denotes a power set of $\mathcal{O}$.
- $A$ is a finite nonempty set of attributes, $A = \{a_1, a_2, \dots, a_n\} \subseteq \text{Þ}\mathcal{A}$.
- $R^c = O \times A$ is a set of internal relations.
- $R^i \subseteq A' \times A, A' \sqsubseteq C' \wedge A \sqsubseteq c$ , is a set of input relations, where $C'$ is a set of external concepts, $C' \subseteq \Theta_c$. For convenience, $R^i = A' \times A$ may be simply denoted as $R_i = C' \times c$.
- $R^o \subseteq c \times C'$ is a set of output relations.

Based on the above definition, an object as a concrete instantiation of a given concept can be derived as follows.

**Definition 3.** An *object o*, *o* ∈ Þ*O*, as a derived bottom-level instantiation of a concept that can be mapped onto a concrete entity can be defined as follows:

$$o \triangleq \left(O_o, A_o, R_o^c, R_o^i, R_o^o\right) \qquad (3)$$

where

- *Oo* = ∅.
- *Ao* is a nonempty set of attributes, *Ao* = {*a1, a2, … , an*} ⊆ ÞA .
- $R_o^c = \emptyset$;
- $R_o^i \subseteq C' \times o$ is a set of input relations, where *C'* is a set of external concepts;
- $R_o^o \subseteq o \times C'$ is a set of output relations.

**Definition 4.** For a concept $c(O, A, R^c, R^i, R^o)$, its intension, *c**, is represented by its set of attributes *A*; while its extension, *c*+, is represented by its set of objects *O*, i.e.:

$$\begin{cases} c^* = A \\ c^+ = O \end{cases} \qquad (4)$$

*3.2. Concept Algebra for Knowledge Manipulations*

Concept algebra is an abstract mathematical structure for the formal treatment of concepts and their algebraic relations, operations, and associative rules for composing complex concepts.

**Definition 5.** A *concept algebra CA* on a given semantic environment $\Theta_c$ is a triple, i.e.:

$$CA \triangleq (C, OP, \Theta_c) = \big((O, A, R^c, R^i, R^o), \{\cdot_r, \cdot_c\}, \Theta_c\big) \qquad (5)$$

where *OP* = {•r, •c} are the sets of *relational* and *compositional* operations on abstract concepts.

**Definition 6.** The *relational operations* **•r** in concept algebra encompass 8 comparative operators for manipulating the algebraic relations between concepts, i.e.:

$$\cdot_r \triangleq \{\leftrightarrow, \nleftrightarrow, \prec, \succ, =, \cong, \sim, \triangleq\} \qquad (6)$$

where the relational operators stand for *related, independent, subconcept, superconcept, equivalent, consistent, comparison,* and *definition*, respectively.

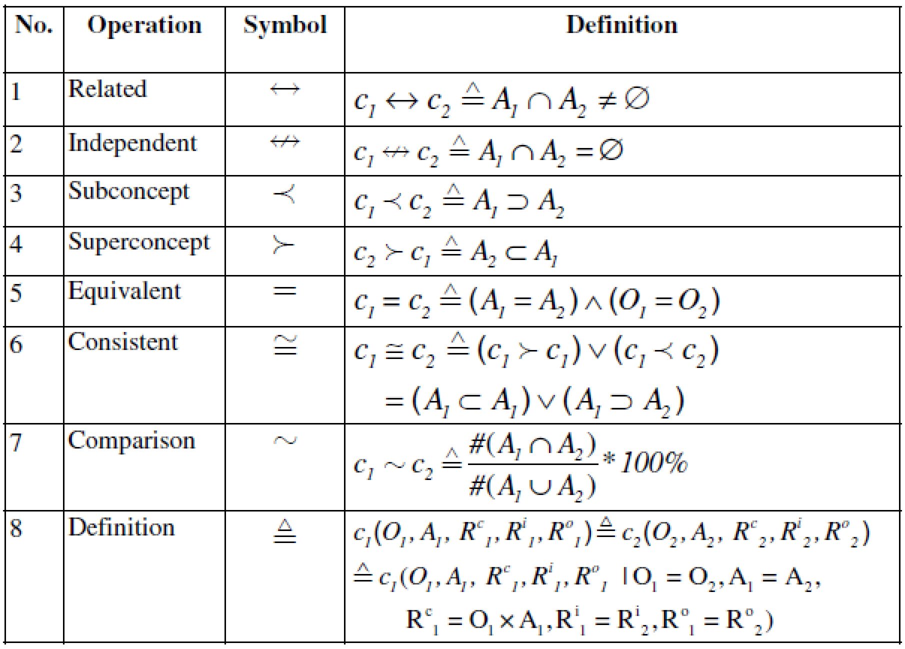
**Definition 7.** The *compositional operations* **•c** in concept algebra encompasses 9 associative operators for manipulating the algebraic compositions among concepts, i.e.:

$$\cdot_c \triangleq \{\Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \uplus, \pitchfork, \Lleftarrow, \vdash, \longmapsto\} \qquad (7)$$

where the compositional operators stand for *inheritance, tailoring, extension, substitute, composition*, *decomposition*, *aggregation*, *specification*, and *instantiation*, respectively.

### *3.2.1. Concept Relational Operations*

In concept algebra, a set of 8 relational operations on abstract concepts, such as subconcept, super-concept, related-concepts, independent-concepts, equivalent-concepts, consistent-concepts, concept comparison, and definition are modeled in Table 1.



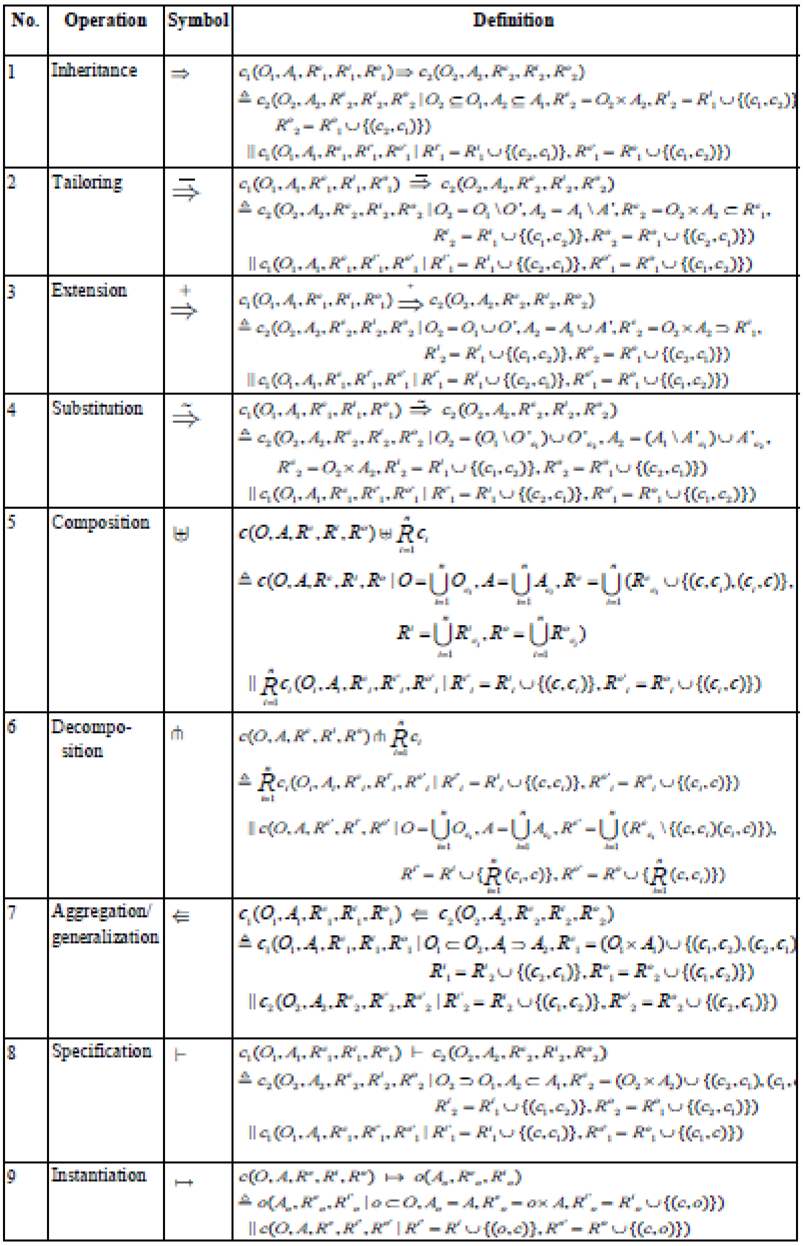
| No. | Operation | Symbol | Definition |
|---|---|---|---|
| 1 | Related | $\leftrightarrow$ | $c_1 \leftrightarrow c_2 \triangleq A_1 \cap A_2 \neq \varnothing$ |
| 2 | Independent | $\nleftrightarrow$ | $c_1 \nleftrightarrow c_2 \triangleq A_1 \cap A_2 = \varnothing$ |
| 3 | Subconcept | $\prec$ | $c_1 \prec c_2 \triangleq A_1 \supset A_2$ |
| 4 | Superconcept | $\succ$ | $c_2 \succ c_1 \triangleq A_2 \subset A_1$ |
| 5 | Equivalent | $=$ | $c_1 = c_2 \triangleq (A_1 = A_2) \wedge (O_1 = O_2)$ |
| 6 | Consistent | $\cong$ | $c_1 \cong c_2 \triangleq (c_1 \succ c_1) \vee (c_1 \prec c_2)$<br>$= (A_1 \subset A_1) \vee (A_1 \supset A_2)$ |
| 7 | Comparison | $\sim$ | $c_1 \sim c_2 \triangleq \frac{\#(A_1 \cap A_2)}{\#(A_1 \cup A_2)} * 100\%$ |
| 8 | Definition | $\triangleq$ | $c_1(O_1, A_1, R^c{}_1, R^i{}_1, R^o{}_1) \triangleq c_2(O_2, A_2, R^c{}_2, R^i{}_2, R^o{}_2)$<br>$\triangleq c_1(O_1, A_1, R^c{}_1, R^i{}_1, R^o{}_1 \mid \mathrm{O}_1 = \mathrm{O}_2, \mathrm{A}_1 = \mathrm{A}_2,$<br>$\mathrm{R}^c{}_1 = \mathrm{O}_1 \times \mathrm{A}_1, \mathrm{R}^i{}_1 = \mathrm{R}^i{}_2, \mathrm{R}^o{}_1 = \mathrm{R}^o{}_2)$ |

**Table 1:** Relational Operations of Concept Algebra

The *related concepts* are a pair of concepts that share some common attributes in their intensions; while the independent concepts are two concepts that their intensions are disjoint. The *subconcept* of a given concept is a concept that its intension is a superset; while a *superconcept* over a concept is a concept that its intension is a subset of the subconcept. The *equivalent concepts* are two concepts that both of their intensions and extensions are identical. The *consistent concepts* are two concepts with a relation of being either a sub- or super-concept. A *comparison* between two concepts is an operation that determines the equivalency or similarity level of their intensions. Concept comparison is implemented according to the definition in Table 1, where the range of similarity between two concepts is between 0 to 100%, in which 0% means no similarity and 100% means a full similarity between two given concepts. *Concept definition* is an association between two concepts where they are equivalent.

### *3.2.2. Concept Compositional Operations*

In concept algebra, nine concept associations have been defined as shown in Table 2 encompassing *inheritance, tailoring, extension, substitute, composition, decomposition, aggregation, specification,* and *instantiation* (Wang, 2009).



| No. | Operation | Symbol | Definition |
|---|---|---|---|
| 1 | Inheritance | $\Rightarrow$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \Rightarrow c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_2(O_2,A_2,R^c_2,R^i_2,R^o_2 \mid O_2 \subseteq O_1, A_2 \subseteq A_1, R^c_2 = O_2 \times A_2, R^i_2 = R^i_1 \cup \{(c_1,c_2)\}$<br>$R^o_2 = R^o_1 \cup \{(c_2,c_1)\})$<br>$\Vert c_1(O_1,A_1,R^c_1,R^{i'}_1,R^{o'}_1 \mid R^{i'}_1 = R^i_1 \cup \{(c_2,c_1)\}, R^{o'}_1 = R^o_1 \cup \{(c_1,c_2)\})$ |
| 2 | Tailoring | $\overset{-}{\Rightarrow}$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \overset{-}{\Rightarrow} c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_2(O_2,A_2,R^c_2,R^i_2,R^o_2 \mid O_2 = O_1 \setminus O', A_2 = A_1 \setminus A', R^c_2 = O_2 \times A_2 \subset R^c_1,$<br>$R^i_2 = R^i_1 \cup \{(c_1,c_2)\}, R^o_2 = R^o_1 \cup \{(c_2,c_1)\})$<br>$\Vert c_1(O_1,A_1,R^c_1,R^{i'}_1,R^{o'}_1 \mid R^{i'}_1 = R^i_1 \cup \{(c_2,c_1)\}, R^{o'}_1 = R^o_1 \cup \{(c_1,c_2)\})$ |
| 3 | Extension | $\overset{+}{\Rightarrow}$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \overset{+}{\Rightarrow} c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_2(O_2,A_2,R^c_2,R^i_2,R^o_2 \mid O_2 = O_1 \cup O', A_2 = A_1 \cup A', R^c_2 = O_2 \times A_2 \supset R^c_1,$<br>$R^i_2 = R^i_1 \cup \{(c_1,c_2)\}, R^o_2 = R^o_1 \cup \{(c_2,c_1)\})$<br>$\Vert c_1(O_1,A_1,R^c_1,R^{i'}_1,R^{o'}_1 \mid R^{i'}_1 = R^i_1 \cup \{(c_2,c_1)\}, R^{o'}_1 = R^o_1 \cup \{(c_1,c_2)\})$ |
| 4 | Substitution | $\overset{\sim}{\Rightarrow}$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \overset{\sim}{\Rightarrow} c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_2(O_2,A_2,R^c_2,R^i_2,R^o_2 \mid O_2 = (O_1 \setminus O'_{c_1}) \cup O'_{c_2}, A_2 = (A_1 \setminus A'_{c_1}) \cup A'_{c_2},$<br>$R^c_2 = O_2 \times A_2, R^i_2 = R^i_1 \cup \{(c_1,c_2)\}, R^o_2 = R^o_1 \cup \{(c_2,c_1)\})$<br>$\Vert c_1(O_1,A_1,R^c_1,R^{i'}_1,R^{o'}_1 \mid R^{i'}_1 = R^i_1 \cup \{(c_2,c_1)\}, R^{o'}_1 = R^o_1 \cup \{(c_1,c_2)\})$ |
| 5 | Composition | $\uplus$ | $c(O,A,R^c,R^i,R^o) \uplus \overset{n}{\underset{i=1}{R}} c_i$<br>$\triangleq c(O,A,R^c,R^i,R^o \mid O = \bigcup_{i=1}^{n} O_{c_i}, A = \bigcup_{i=1}^{n} A_{c_i}, R^c = \bigcup_{i=1}^{n} (R^c_{c_i} \cup \{(c,c_i),(c_i,c)\},$<br>$R^i = \bigcup_{i=1}^{n} R^i_{c_i}, R^o = \bigcup_{i=1}^{n} R^o_{c_i})$<br>$\Vert \overset{n}{\underset{i=1}{R}} c_i(O_i,A_i,R^c_i,R^{i'}_i,R^{o'}_i \mid R^{i'}_i = R^i_i \cup \{(c,c_i)\}, R^{o'}_i = R^o_i \cup \{(c_i,c)\})$ |
| 6 | Decompo-sition | $\pitchfork$ | $c(O,A,R^c,R^i,R^o) \pitchfork \overset{n}{\underset{i=1}{R}} c_i$<br>$\triangleq \overset{n}{\underset{i=1}{R}} c_i(O_i,A_i,R^c_i,R^{i'}_i,R^{o'}_i \mid R^{i'}_i = R^i_i \cup \{(c,c_i)\}, R^{o'}_i = R^o_i \cup \{(c_i,c)\})$<br>$\Vert c(O,A,R^{c'},R^{i'},R^{o'} \mid O = \bigcup_{i=1}^{n} O_{c_i}, A = \bigcup_{i=1}^{n} A_{c_i}, R^{c'} = \bigcup_{i=1}^{n} (R^c_{c_i} \setminus \{(c,c_i)(c_i,c)\}),$<br>$R^{i'} = R^i \cup \{\overset{n}{\underset{i=1}{R}} (c_i,c)\}, R^{o'} = R^o \cup \{\overset{n}{\underset{i=1}{R}} (c,c_i)\})$ |
| 7 | Aggregation/ generalization | $\Leftarrow$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \Leftarrow c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_1(O_1,A_1,R^c_1,R^i_1,R^o_1 \mid O_1 \subset O_2, A_1 \supset A_2, R^c_1 = (O_1 \times A_1) \cup \{(c_1,c_2),(c_2,c_1)$<br>$R^i_1 = R^i_2 \cup \{(c_2,c_1)\}, R^o_1 = R^o_2 \cup \{(c_1,c_2)\})$<br>$\Vert c_2(O_2,A_2,R^c_2,R^{i'}_2,R^{o'}_2 \mid R^{i'}_2 = R^i_2 \cup \{(c_1,c_2)\}, R^{o'}_2 = R^o_2 \cup \{(c_2,c_1)\})$ |
| 8 | Specification | $\vdash$ | $c_1(O_1,A_1,R^c_1,R^i_1,R^o_1) \vdash c_2(O_2,A_2,R^c_2,R^i_2,R^o_2)$<br>$\triangleq c_2(O_2,A_2,R^c_2,R^i_2,R^o_2 \mid O_2 \supset O_1, A_2 \subset A_1, R^c_2 = (O_2 \times A_2) \cup \{(c_2,c_1),(c_1,$<br>$R^i_2 = R^i_1 \cup \{(c_1,c_2)\}, R^o_2 = R^o_1 \cup \{(c_2,c_1)\})$<br>$\Vert c_1(O_1,A_1,R^c_1,R^{i'}_1,R^{o'}_1 \mid R^{i'}_1 = R^i_1 \cup \{(c,c_1)\}, R^{o'}_1 = R^o_1 \cup \{(c_1,c)\})$ |
| 9 | Instantiation | $\mapsto$ | $c(O,A,R^c,R^i,R^o) \mapsto o(A_o,R^c_o,R^i_o)$<br>$\triangleq o(A_o,R^c_o,R^{i'}_o \mid o \subset O, A_o = A, R^c_o = o \times A, R^{i'}_o = R^i_o \cup \{(c,o)\})$<br>$\Vert c(O,A,R^c,R^{i'},R^{o'} \mid R^{i'} = R^i \cup \{(o,c)\}, R^{o'} = R^o \cup \{(c,o)\})$ |

**Table 2:** Compositional Operations of Concept Algebra

*Concept inheritance* is a concept association that indicates one concept derived from another concept. Multiple inheritances are an associative operation where a concept is derived from multiple concepts. *Concept tailoring* is a special inheritance operation on concepts that reduces some inherited attributes or objects in the derived concept. *Concept extension* is an associative operation that carries out a special concept inheritance with the introduction of additional attributes and/or objects in the derived concept. *Concept substitution* is an inheritance operation that results in the replacement or overload of attributes and/or objects by locally defined ones. *Concept composition* is a associative operation that integrates multiple concepts in order to form a new complex one. *Concept aggregation* is an associative operation that assembles of complex concept by using components provided from those of multiple concepts. *Concept specification* is an associative operation that refines a concept by another sub-concept with more specific and precise attributes. *Concept instantiation* is a special associative operation that derives an object, or instance, on the basis of the inherited concept.

The mathematical model of knowledge can be set as a concept network, where concept algebra is applied as a set of rules for knowledge composition. We have to set to more definitions.

**Definition 8.** *A generic knowledge K is an n-nary relation* $\mathfrak{R}$ *among a set of n concepts and the entire set of concepts C, i.e.:*

$$K = \mathcal{R}: (X_{i=1}^{n} C_i \longrightarrow C) \qquad (8)$$

where $\biguplus_{i=1}^{n} C_i = C$ , and $\mathcal{R} = \cdot_c = \{\Longrightarrow, \Longrightarrow, \Longrightarrow, \Longrightarrow, \uplus, \pitchfork, \Lleftarrow, \vdash, \longmapsto\}$

**Definition 9.** A *concept network* CN is a hierarchical network of concepts interlinked by the set of nine composing rules $\mathfrak{R}$ concept algebra, i.e.:

$$CN = \mathcal{R}: X_{i=1}^{n} C_1 \longrightarrow X_{j=1}^{n} C_j \qquad (9)$$

**Theorem 1.** In a concept network CN, the abstract levels of concepts $\ell_c$ form a partial order of a series of superconcepts, i.e.:

$$\ell_c = (\phi \preccurlyeq c_1 \preccurlyeq c_2 \preccurlyeq \cdots \preccurlyeq c_n \preccurlyeq \cdots \preccurlyeq \Omega) \qquad (10)$$

where ∅ is the empty concept ∅ = (⊥, ⊥), and Ω the universal concept, Ω = (O, A).

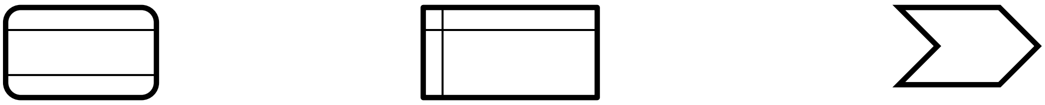
## 3. KAMET II CML: The conceptual modeling language

KAMET II comes from Knowledge Acquisition Methodology, it is a conceptual modeling language (CML) primarily aimed to solve diagnosis problems. It is a high user friendly language based on visual structures and symbols, which can be used to represent concepts and the relation between them. KAMET II is quite intuitive having the advantage that the user does not take so long to get familiar with it.

KAMET II can represent complex knowledge through intuitive tools compatible with diagnosis problems knowledge representation. However, we must take into account that languages that are more expressive are likely to require more complex logic and algorithms to construct equivalent inference. KAMET II is expressive enough to have flexibility when modeling without adding this kind of difficulties. It is important to add that the content of the models depends on the good judgment and considerations of the knowledge engineers (KE), which is the personal in charge to put together all the knowledge tended to be represented. KAMET II can rectify the correctness of the models inserted according to its semantic rules but it cannot assure anything about the liability of the content.

### *2.1. The KAMET II CML assumptions*

KAMET II CML has three levels of abstraction. The first one corresponds to *structural constructors* and *structural components*. The structural constructors are used to describe the problem itself (Figure 1) and the structural components, on the other hand, are used to establish the possible solutions to the problem (Figure 2).



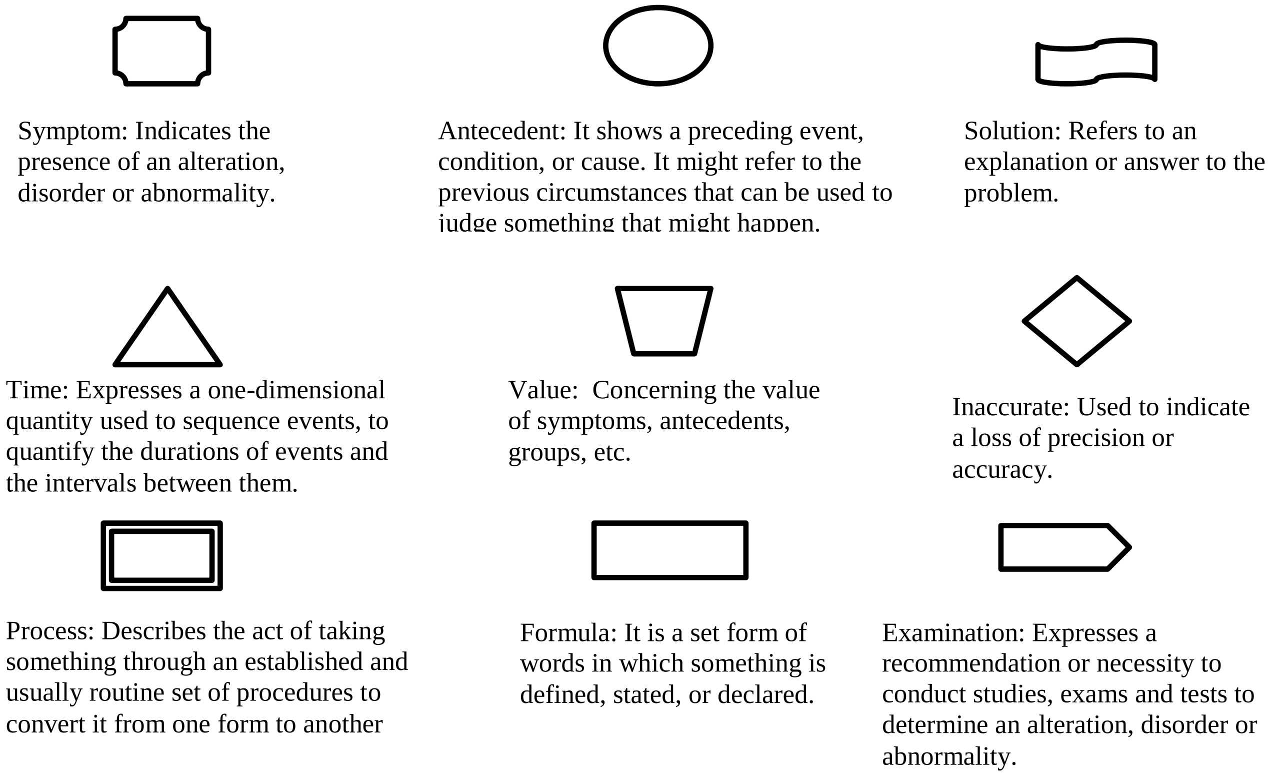
Problem: It refers to a situation, condition, or issue that is still unresolved.

Classification: A characteristic inference structure that systematically relates data to a preenumerated set of solutions by abstraction, heuristic association, and refinement.

Subdivision: It is the act of dividing a problem into small pieces that are easier to solve.

**Figure 1:** Structural constructors

The structural components are three: *problem*, *classification* and *subdivision*. They are used in the terminal nodes how it is defined later. The structural components are nine: *symptom*, *antecedent*, *solution*, *time*, *value*, *inaccurate*, *process*, *formula* and *examination*, which can be mixed to define the antecedents of a problem by mentioning one way to use them.



Symptom: Indicates the presence of an alteration, disorder or abnormality.

Antecedent: It shows a preceding event, condition, or cause. It might refer to the previous circumstances that can be used to judge something that might happen.

Solution: Refers to an explanation or answer to the problem.

Time: Expresses a one-dimensional quantity used to sequence events, to quantify the durations of events and the intervals between them.

Value: Concerning the value of symptoms, antecedents, groups, etc.

Inaccurate: Used to indicate a loss of precision or accuracy.

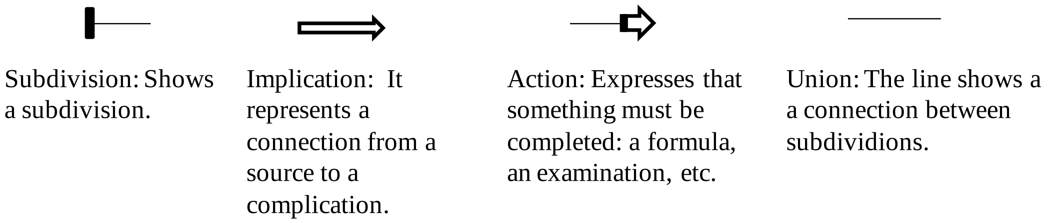
Process: Describes the act of taking something through an established and usually routine set of procedures to convert it from one form to another

Formula: It is a set form of words in which something is defined, stated, or declared.

Examination: Expresses a recommendation or necessity to conduct studies, exams and tests to determine an alteration, disorder or abnormality.

**Figure 2:** Structural components

The second level of abstraction corresponds to *nodes* (N) and *composition rules* (CR). Nodes are built using structural constructors and structural components. We distinguish between three different types of nodes: *initial*, *intermediate* and *terminal*. Composition rules (Figure 3), for their part, are the ones that permit the appropriate combination of nodes.



Subdivision: Shows a subdivision.

Implication: It represents a connection from a source to a complication.

Action: Expresses that something must be completed: a formula, an examination, etc.

Union: The line shows a a connection between subdividions.

**Figure 3:** Composition rules

The third level of abstraction corresponds to the global model. It consists of at least one initial node, any number of intermediate nodes, and one or more terminal nodes. A global model should represent the knowledge acquired from multiple knowledge sources in a specific knowledge domain.

### *2.2. Diagrammatic convention for KAMET II CML*

A diagrammatic convention is a chart, graph or outline designed to demonstrate or explain how something works or to clarify the relationship between the parts of a whole. Here we define the diagrammatic conventions:

DG1. The structural constructors and structural components can be *named* using a numerical or linguistic label (Figure 4). The use of names accelerates and facilitates the construction of models.

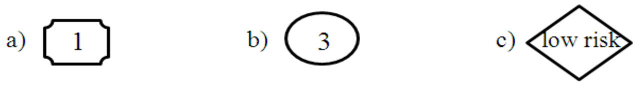


**Figure 4:** Names of structural constructors and structural components: a) symptom: 1; b) antecedent: 3; c) inaccurate: low risk.

DG2. The *indicator* is used to set up the number of elements that must be present in either a structural component or group. It is represented with a square (Figure 5) and is located in the upper right-hand corner of the group or the structural component. An indicator is named in three different ways: an *n* is used to express the exact number of elements that must be present, an *n*+ is used to indicate that at least *n* elements must be present, and an *n,m* is used to show the minimum and maximum number of elements that must be present, where *n* and *m* are integer values, and *m*>*n*.

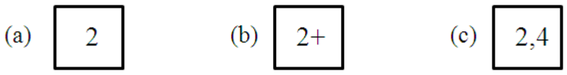


**Figure 5:** Indicators: a) n; b) n+; c) n,m

DG3. A *chain* (Figure 6) is defined as the link of two or more symptoms, antecedents, and/or groups (DG4). The order of the link is irrelevant.

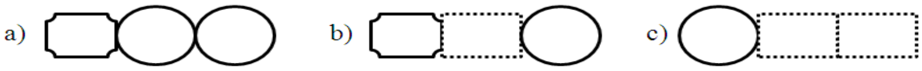


**Figure 6:** Chains: a) symptom and two antecedents; b) symptom, group and antecedent; c) antecedent and two groups

DG4. A *group* (Figure 7) is defined as a special chain. The linked elements have times and/or values in common, or are related among each other through an indicator. The group concept is recursive.

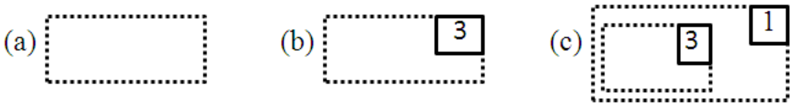


**Figure 7:** Groups: a) group; b) group with an indicator n; c) recursive group

DG5. *Assignment* (Figure 8) is defined as the process of labeling a node. The objective of the assignment is to be able to reuse the node in any other part of the model without having to redefine it. It allows reusing a complete node not only in form but also in content. Reusing is a universal principle of coping with complexity and to avoid redesigning or redeveloping parts of a product, which already exist. The assignment provides greater flexibility in modeling.

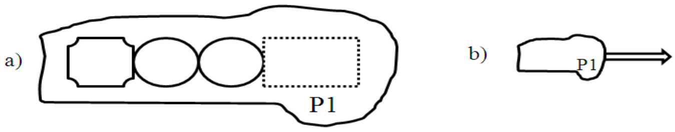


**Figure 8:** Names assignation: a) labeling a node; and b) using a labeled node

*2.2.1. Postulates*

The postulate or axiom is a proposition in logic that is not proved or demonstrated but is considered to be either self-evident or assumed to be true as a basis for reasoning. Its truth is taken for granted, and serves as a starting point for deducing and inferring other propositions. Following are the postulates of the method:

P1. The structural component *time* (Figure 9) should always be placed to the right of a group, problem, subdivision, antecedent, symptom, etc.

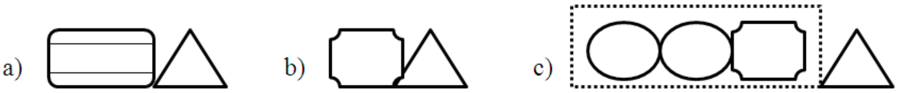


**Figure 9:** Time: a) problem with time; b) symptom with time; and c) group with time

P2. The structural component *value* (Figure 10) is always placed above a symptom, antecedent or group. The value component can make use of an indicator.

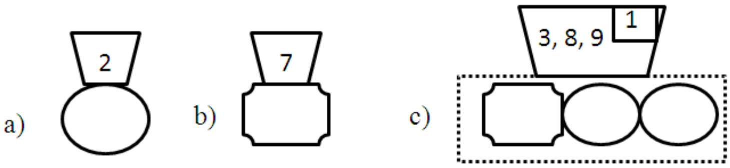


**Figure 10:** Value: a) antecedent with a value; b) symptom with a value; and c) group with three values and one indicator. It means only one of these values must be present.

P3. The solution component is only related to structural constructors.

P4. There are three types of nodes: initial (I), intermediate (M) and terminal (T).

P5. The *nodes* are related using composition rules. The following relationships are possible: initial with terminal, initial with intermediate, intermediate with intermediate, and intermediate with terminal.

P6. An *initial node* represents a symptom, antecedent, group, or chain. It is used to describe a part of the problem. It does not have input flow and can have more than one output flow.

P7. The *intermediate node* is used to describe an intermediate part of the problem. It may have one or more further inflows and one or more output flows.

P8. A *terminal node* represents a structural constructor. It has one or more input flows. The output flow is only used to show possible solutions.

P9. The initial and intermediate nodes can be grouped together, without losing their properties or functions, into molecular nodes. These nodes, in turn, will act as a node in their own right. The molecular nodes are formed through conjunctions or disjunctions (Figure 12).

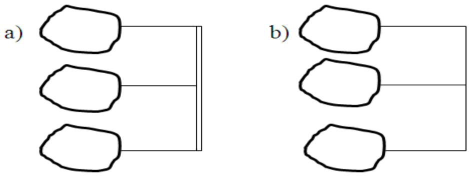


**Figure 12:** Molecular nodes: a) conjunction; and b) disjunction

P10. The *composition rules* are used mainly to relate the different nodes and the structural components with the solution component.

*2.2.2. A simple example of modeling*

(Figure 13) provides a simple example of a model built with KAMET II in order to illustrate the method sketched in the previous section. The example models the problem of deterioration due to aggressive water (P2.4). The antecedent A27 represents the fact that the structure carries a water current. The dotted rectangle in this figure has an indicator that at least one symptom (S16 and S17) inside the rectangle must be presented. S14 indicates dissolving of the paste exposing the aggregates, S16 represents the presence of holes on the surface and S17 the presence of grains of sand on the surface of the concrete. The presence of the symptoms and antecedents mentioned above let us conclude the existence of problem P2.4 with a probability of p, showed in a rounded rectangle.

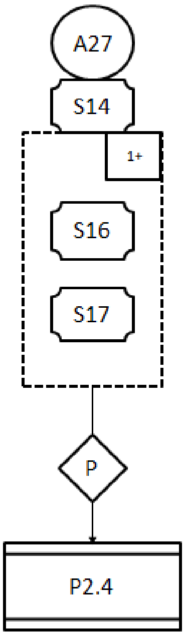


**Figure 13:** Simple electrical diagnosis

## 4. The KAMET II CML consistency and transparency

Formalization refers to an explicit syntax and unambiguous semantics for the term used, giving a definite form or shape to conceptual modeling language. It allows for such nice features as consistency and transparency. We use concept algebra to formalize KAMET II.

The second and third level of abstraction of KAMET II, that means its global model and nodes are analog and consistent to the abstract concepts of Concept Algebra, so do the structural constructors and structural components with concepts and attributes. In order to probe the consistency of KAMET II some conventions have to be done:

Let use definition 9 to set a CN with abstract levels of concepts $\ell_c$ forming a partial order of a series of superconcept using equation (10) where each level is built with certain concepts from the KAMET II model by

$$\ell_c = (\phi \preccurlyeq c_1 \preccurlyeq c_2 \preccurlyeq \cdots \preccurlyeq c_n \preccurlyeq \cdots \preccurlyeq \Omega)$$

where $c_n$ define concepts from the domain of the knowledge represented by the KAMET II diagram or context $\Theta_c \triangleq (\mathcal{O}, \mathcal{A}, \mathcal{R})$ related to the same diagram.

C1. Each initial, intermediate and terminal node that consists of more than one member or object has to be assigned with a label or identifier so it can be seen like a hole by using the DG5, assignment rule. The labels in the visual representation have to be set, for example, W1.1, always and W before a number. It is necessary to define the nodes as a complete abstract concept and that it is easier to manipulate them and reuse them. Nodes with just one member keep the same structure.

C2. The structural constructors represent abstract concepts so do it some of the structural components: *symptom*, *antecedent*, *solution*, *process*, *formula* and *examination*. The structural components that refer to *time*, *value* and *inaccurate* are represented as attributes of concepts. From equation (5), C is the set of concepts that can be taken from the KAMET II according to this convention and the set of attributes related to A in the same equation are set as,

$$A = \{a_1 \ = time, a_2 = value, a_3 = innacurance \}$$

C3. Each KAMET II model can be represented like hierarchical relations between concepts. Each node of the KAMET II model is an abstract concept by itself. Using definition 1 and definition 2, we then set that a semantic environment context $\Theta_c$ exists and it represent the set of all objects, attributes and their relations involved in the model by $\Theta_c \triangleq (\mathcal{O}, \mathcal{A}, \mathcal{R})$ which is equation (1). Also with equation (2) we denoted de definition of an abstract concept, according to this equation we can set that initial nodes, represented by $c_{i\_n}$ when taken as concept, have no input relations but internal an output relations.

$$c_{i\_n} \triangleq \left(O, A, R^c, R^i, R^o\right)$$

where we set the same definition of $O, A, R^c, R^o$ in equation (1) but $R^i$, which is $R^i = \emptyset$. For intermediate nodes, $c_{m_n}$we define

$$c_{m\_n} \triangleq \left(O, A, R^c, R^i, R^o\right)$$

with same definition of $O, A, R^i, R^c, R^o$ in equation (1). For terminal nodes $c_{t\_n}$ we keep the same definition of $O, A, R^i, R^c, R^o$ in equation (1) but$R^o$, which is $R^o = \emptyset$, which means that terminal nodes do not have output relations.

$$c_{t\_n} \triangleq \left(O, A, R^c, R^i, R^o\right)$$

The $n$ in the sub index of the notation above should be substituted for the label of the node we are interested to define.

C4. It is illegal that any component during the inference chain gets to itself indirectly, that means cycles are not allowed. It is right to use in the same model instance of the same labeled block but they cannot be related by superconcept or subconcept relation. One concept cannot be composed by itself directly or indirectly.

Defining,

$$c_{m\_w1.1} \triangleq \left(O_{m\_w1.1}, A_{m\_w1.1}, {R^c}_{m\_w1.1}, {R^i}_{m\_w1.1}, {R^o}_{m\_w1.1}\right)$$

and

$$c_{m\_w1.2} \triangleq \left(O_{m\_w1.2}, A_{m\_w1.2}, {R^c}_{m\_w1.2}, {R^i}_{m\_w1.2}, {R^o}_{m\_w1.2}\right)$$

Setting that $c_{m\_w1.1} \succ c_{m_{w1.2}} \succ \cdots \succ c_{m_{w1.n}}$, then $c_{m\_w1.n} \succ c_{m_{w1.n-1}} \succ \cdots \succ c_{m_{w1.1}}$ cannot be true for this indirect relation nor any level relation.

C5. Due to the fact that the mean structure of the KAMET II models are AND, OR, AT LEAST, MIN/MAX and the EXACT number of elements that has to be presented, in order to deduce that a problem is happening, the KAMET II model can be seen like a hierarchical composition diagram, where a problem is built on the presence of elements of lower hierarchy. It can be created different subconcept made out the permutation of the relations said before and set the similitude between each new subconcept.

Taking the KAMET II from (figure 13), simple electrical diagnosis, we can start by turning the diagram in to a logic transformation model which the result would be the logical diagram (figure 14). Let notice that model from (figure 13) owns 3 nodes: one initial, which includes all the symptoms and antecedents, the inter medial which is just the probability and the final node which denotes the problem. For convention de probability is treated as an attribute of the subconcepts W1.1 which is the abstract representation of the initial node.

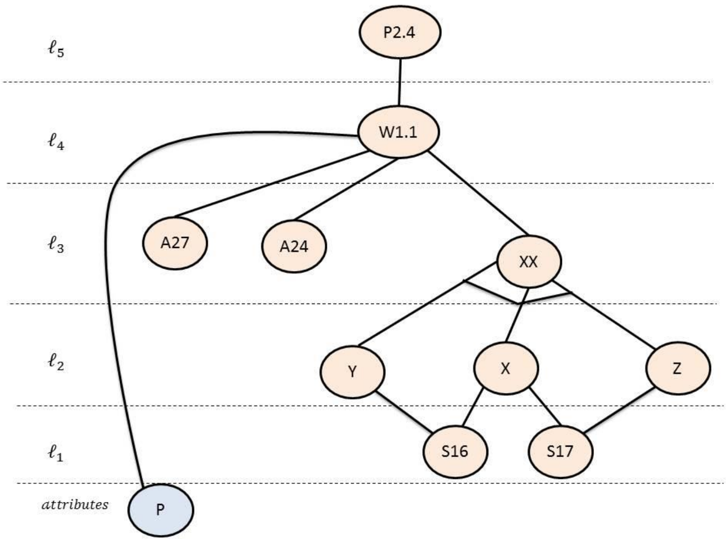


**Figure 14:** Simple electrical diagnosis represented in AND / OR diagram.

In (figure 14) the junctions united by a line indicates OR relation, without this line means AND. Due to represent all the concepts that can lead to the same problem represented by blocks, that means when we have at least n quantity of elements, we can clearly see that there are different set of subconcepts that integrate a concept that has the same probability of producing the subconcept XX, that was set an extra node which has the appropriated AND / OR relations to show all the possible combinations.

C6. We are working with concept, subconcepts and superconcepts according to the hierarchical relation between the components of the original diagram made in KAMET II. The highest level is made out of the final nodes; they don't have out relations how it was established before. The other levels are built according of the flow of the diagram; the lowest level would be made out of the initial nodes and the levels in the middle by intermediate nodes. (figure 14 ) shows those levels,

$$\ell_n = \bigcup_{n=1}^{N} c_n \subseteq C \qquad (11)$$

,where N is the number of total concepts in C, (11) define the concepts that are part of a level like a subset of concepts in C.

C7. The relation between concepts that integrate the same superconcept is evident. Some concepts share subconcepts, so they can be made out of the composition, decomposition, aggregation, substitution and other concept algebra operations. The relational operations defined in section 3.2.1 apply to the KAMET II diagram objects, so do the compositional operations of section 3.2.2

From the example of (figure 14) we can observe that, P2.4 is related to W1.1 and that W1.1 has an attribute called P, which represents de probability of having problem P2.4 when W1.1 is set. In order to have W1.1, A27, S14 and XX has to be observed and for having XX at least one of the two symptoms S16 and S17 (X) has to be presented.

$$c_{p2.4} \leftrightarrow c_{w1.1}$$

$$c_{w1.1} \succcurlyeq c_{a27,}c_{s14}, c_{xx}$$

,where $c_{w1.1} \uplus c_{a27,}c_{s14}, c_{xx}$ ,which means $c_{w1.1}$ is the composition of $c_{a27,}c_{s14}, c_{xx}$ and

$$c_{xx} \succcurlyeq c_{y,}c_{z}, c_{x}$$

,where $c_{x} \uplus c_{y,}c_{z}$ , $c_{x}$ is the composition of $c_{y,}c_{z}$. About the attribute P, we can set, that this belongs to $c_{w1.1}$ and not to the individual parts that conforms it. The probability of a node is the probability of that whole node and all the configurations that can be set has the same probability of causing the problem pointed.

We represent the OR junction for X as a set of three concepts that are related, $c_{xx}$ is the superconcept related to $c_{y,}c_{z}, c_{x}$ . $c_{xx}$ is true when one of $c_{y,}c_{z}, c_{x}$ is also true, so they are equivalent.

### *4.1 KAMET II Integrated Development Environment (IDE)*

It exist an Integrated Development Environment (IDE) that helps the user to create the models respecting the rules that constraint the visual language, KAMET II. It rectifies the validation and correctness of the knowledge models involved.

The user drags and drops the different elements to create the models. How it was said before, KAMET II IDE can only check the models for correctness related to the visual language rules. The veracity of the information contained in the models depends only on the experts of the topic.

## 5. Future work

Setting that KAMET II is transparent and consistent by using Concept Algebra, we are also able to translate KAMET II models to different ontology representation languages due to expand the scope of sharing and running the models. We are working on translating KAMET II diagram to different conceptual languages like OWL, OIL, etc, and also to be used in different methodologies for knowledge representation like Protégé 2000. It is important to make sure the models can be translated and also suitable to ontology representation protocols.

There is also another point of improvement directed to user experience. We are adding some works on symbology to adapt the visual elements to more intuitive objects. That means we want to adapt the symbols we use for the more intuitive ones, for example, someone knows when someone is talking about health by looking the to the symbol of the cross with the snake around it.

Timing and probability can be attributes of all concepts. The probabilities of happening marked with a round square are taken as attributes of a concept according to the scope of each element. The result concept network is not deterministic; the existence of an element has a probability of accuracy. These logic relations are used in the inference machine are used to deduce when a problem is happening. The probabilities could be transformed to a Markov Chain representation, where the relations between nodes are represented by the transitions between states with a probability defined by user.

## 6. Conclusions

KAMET II is a complete visual language with high usability devised to acquire and organize knowledge from different sources in a very intuitive way. Similar recent languages are due to machine interpretation and use of knowledge without considering the users interaction. We demonstrate the advantages of using KAMET II which includes user friendly interface, intuitive tools and completeness probed using concept algebra. KAMET II is also compatible with different Ontologies representations like OWL and OIL and is compatible too with Methodologies like Protégé 2000.

## Acknowledgement

I want to thank all the people involved in this project.

## References

Antonious, G., and van Harmelen, F. (2004), *Web ontology Language: OWL*, In S. Staab et al. (eds.), Handbook on Ontologies, pp 67-92, Springer-Verlag Berlin Heidelberg.

Byrd, T.A. (1995). “Expert Systems Implementation: Interviews with Knowledge Engineers.” Industrial Management and Data Systems, Vol. 95, No. 10.

Cairó, O., and Guardati, S. (2012), “The KAMET II methodology: Knowledge acquisition, knowledge modeling and knowledge generation”, *Expert Systems with Applications*, Vol. 39, No. 9, pp 8108-8114.

Cairó, O., and Alvarez, J. (2005), “KAMET: An open diagnosis knowledge acquisition methodology”, *Journal of Computational Methods in Sciences and Engineering*, Vol. 5, No. 1, pp 145-158.

Cairó, O. (1998), “KAMET: A Comprehensive Methodology for Knowledge Acquisition from Multiple Knowledge Sources”, *Expert Systems with Applications*, Vol. 14, pp 1-16.

Davis, R. (1993). “Interactive Transfer of Expertise: Acquisition of New Inference Rules.” In B.G. Buchanan and D. Wilkins (eds.). Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems. San Mateo, CA: Morgan Kaufmann.

Fensel, D., J. Angele, and R. Struder. (1998, July/August). “The Knowledge Acquisition and Representation Language KARL.” IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 4.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D., and Patel-Schneider, P. (2001). “OIL: an ontology infrastructure for the Semantic Web”, *Intelligent Systems*, Vol. 16, No. 2, pp 38-45.

Musen, M.A., L.M. Fagan, D.M. Combs, and E.H. Shortliffe. (1999, August). “Use of a Domain Model to Drive an Interactive Knowledge-Editing Tool.” International Journal of Human–Computer Studies.

Lassila, O., and Swick, R. (1999), *Resource Description Framework (RDF) Model and Syntax Specification*, http://www.w3.org/TR/REC-rdf-syntac/

Rincon, M., M. Bachiller, and J. Mira. (2005, July). Knowledge Modelling for the Image Understanding Task as a Design Task.” Expert Systems with Applications, Vol. 29, No. 1.

Wagner, W.P., J.R. Otto, and Q.B. Chung. (2003). "The Impact of Problem Domains and Knowledge Acquisition Techniques: A Content Analysis of P/OM Expert System Case Studies." Expert Systems with Applications, Vol. 24, pp. 79–86.

Wang, Y., Tian Y. and Hu, K., (2009), *"A Knowledge Representation Tool for Autonomous Machine Learning Based on Concept Algebra"*, Trans. On Computer Science pp 143-160.

Wang, Y., (2008) *"On Concept Algebra: A Denotational Mathematical Structure for Knowledge and Software Modeling"*, Journal of Cognitive Informatics and Natural Intelligence.