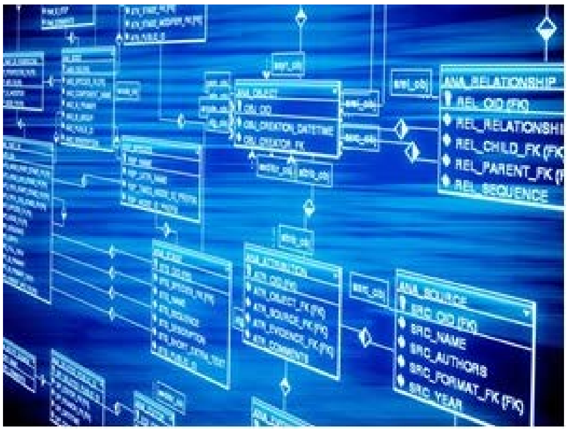
Chimica & Materiali

AN UML CLASS DIAGRAM AS *META*MODEL OF THE KNOWLEDGE DOMINION OF THE ENGINEERING "UNIT OPERATIONS"



Gianni Grasso
Scuola di Scienze Agrarie Unibas (PZ)
(docente esterno)
Gennaro Bufalo
Inail Settore Ricerca Napoli
g.bufalo@inail.it

UML Class diagrams are widely used for purposes such as meta-modeling and database and ontology engineering. This paper opens with a concise presentation of the class diagram model, which is followed by an applicative illustration concerning the "Engineering Unit Operation" knowledge. A taxonomy diagram of this sphere of knowledge, showing its concepts hierarchy, is consequently driven. I.e. a visual layout that correlates in a logical theory the physical system description to the main classes of its equations. The latters are scattered in the field of the applied Fluid Dynamics, Thermodynamics, Industrial Chemistry, Food Applied Science and Biotechnology.

***Meta*-modellazione del dominio di conoscenza delle "Operazioni Unitarie" dell'Ingegneria Chimica mediante un diagramma delle classi UML**
Lo strumento rappresentativo del "diagramma delle classi", componente essenziale del linguaggio "UML" (Unified Modeling Language), fornisce un modello standardizzato per visualizzare l'architettura di un sistema formalizzato in classi e loro associazioni. Nel caso considerato il sistema esaminato è il dominio di conoscenza delle "Operazioni Unitarie" dell'Ingegneria Chimica. La tassonomia di tale dominio, costituito da concetti, equazioni e loro relazioni, diventa in tal modo la base suscettibile di una eventuale implementazione software.

Introduction

UML (Unified Modeling Language) *class diagrams* are clear and intuitive visual models in order to specify the static structure of a given system [1]. *Class diagrams* have been utilised to describe partial ontologies in the materials and molecular biology domains [2, 3].

In the present paper, a *class diagram* models the specific knowledge dominion of the so-called Engineering "Unit Operations" of the chemical, bio-technological, pharmaceutical and food industries [4]. The "Unit Operations" form the basis of the technical, multidisciplinary knowledge of the engineers and scientists that work in the cited industrial fields.

They also serve as a primitive framework for knowledge representation in the considered dominion, in the perspective of specifying more detailed ontologies generating implementation artifacts such as software developments or database schemata.

In Fig. 1 there is our proposed *class diagram* in the “Unit Operations” domain, using UML to define the structure and interactions within the considered knowledge system. The sequential numbers superscripts refer to the annexed front-notes, that concisely explain the meanings of the cited terms/symbols or the formal architecture of the cited equations [5].

The superscripted letters on the arrows refer instead to the basic properties of the indicated relationships (see the key at bottom of the Fig. 1). We indeed can recognize in our diagram four basic logic relationships according to the UML standard formalism [6]:

1) unidirectional *association* relationships (continuous line — with an arrow →) and a specified property; “a = << is specified by/with …>> or <<is defined by …>> or <<is associated with …>>”;
2) *dependency* relationships (dashed line ---- with an arrow →) and specific properties such as:
“b = << depends on …>>” or << is based on …>>;
“c = <<refers to …>> or <<presupposes a …>>”;
3) *inheritance* or <<is a kind of …>> relationships (hollow triangle shape ▷) ;
4) *composition* or <<is made of …>> relationships (filled diamond shape ◆).

We can recognize: a) some inheritance taxonomies at two-levels as classification tools (systems, systems surroundings, boundary attributes, system states, system variables and unit structures), b) a completely defined composition (structure, entirely made of parts and connections), c) several indicated compositions, d) and several association and dependencies, duly completed with meanings.

Implementation opportunities

a. Data-base aspects

The resulting concepts structure is strictly hierarchical, so that each node (a *class*, i.e. a concept or a set of objects) has only one superior node (taxonomy structure). Any term of the graph, i.e. any object, could be specified and differentiated in the instances, and types and attributes of any object could be furtherly “deployed” in the standard syntax of the UML class category; e.g. proper class name, attributes/properties, methods/operations, values and multiplicity as instances components. An illustrative example, concerning the “Unit Structure” term is contained in Ref. [7]. Navigability among concepts is further suggested by the arrows or non-arrows associations.

Our diagram, consisting of a set of concepts (terms or atomic types) and hierarchical relationships among them, so configures substantially as a taxonomy approaching a light-weight formal ontology for software specification or for web search engines [8].

b. Cognitive aspects

The preferential direction of the arrows of the associations is, obviously, toward the superior node of the network, i.e. the “system” term. From a cognitive viewpoint, this direction follows a bottom-up deductive process but, reversing this direction, we can start from the “system” concept to the intermediate concepts and finally to the applications, as mathematical (equations, in the right part of the diagram) as in the form of potential exportable knowledge in new fields (in the bottom, at left, part of the diagram). In deductive inference, we recognize the “system theory” and, based on it, we make a prediction of its consequences in the aforementioned technical applications.

This approach will be useful in a modelization strategy that explicates, starting from an unique father model, the real word differentiation; i.e. as an operative *meta*-model covering the entire domain of the subject, putting the acquired knowledge into a coherent structure. In summary a concepts Novak’s map better known as “mind map” [9, 10].

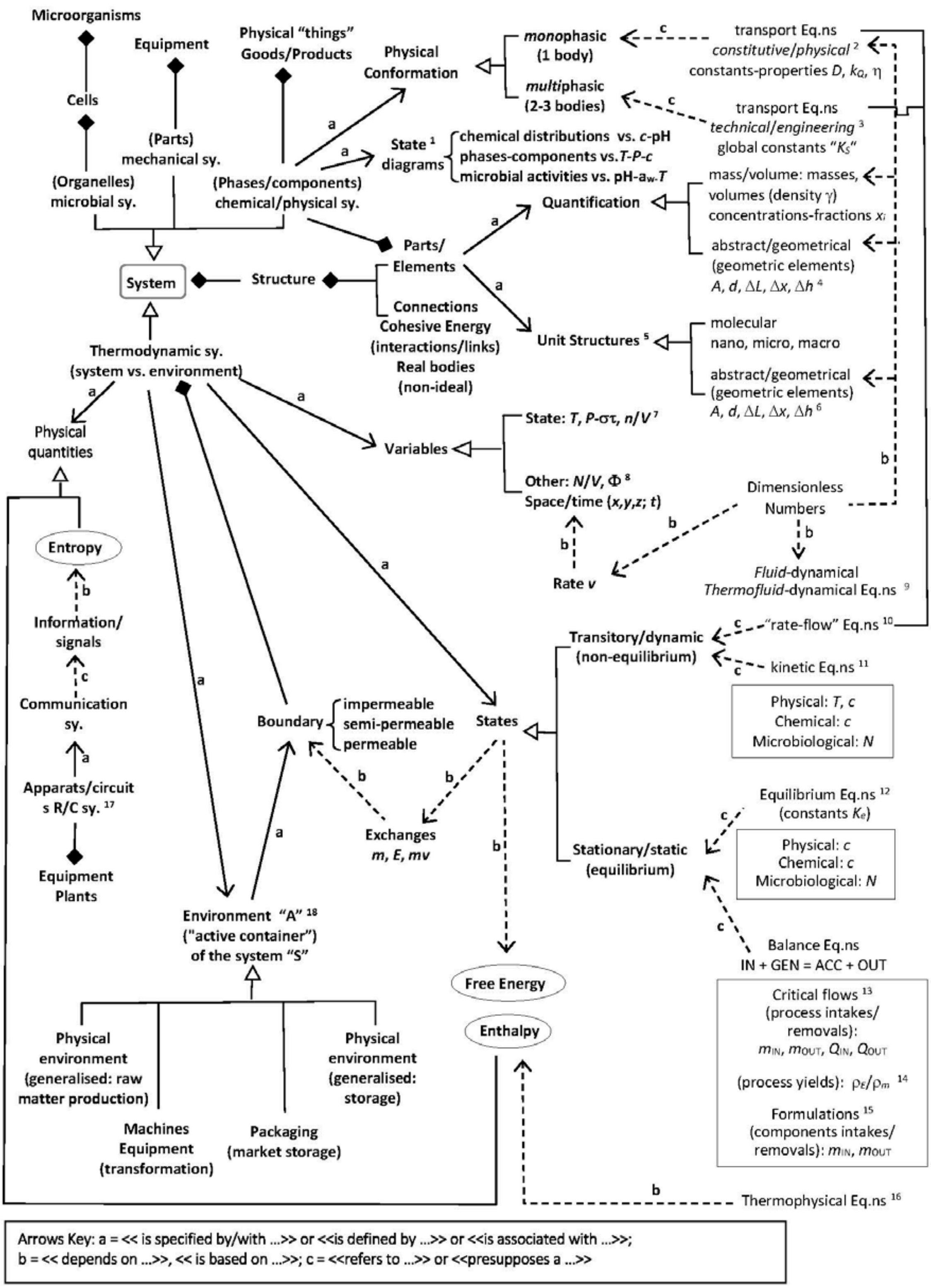


Fig. 1 - UML class diagram representing the taxonomy of the "Unit Operations" equations. Also knowledge map expressed in the UML standard language. System characters on the left, formal models on the right

Notes to figure

1. Diagrams showing the behavior of the structure-classes of the system in response to external stimuli (= actual changes in the system states consequent to changes of the environment state variables; see later notes ≠ 7 and 8). These classes can be molecules in varying pH contexts (chemical species, quantified by their respective concentrations), physical phases, colloids or polymers in process-conditions as intended by Physical Chemistry or Materials Science, or also microorganisms exposed to friend or hostile life-variables. Matter can actually be either unanimated or animated (active, un-active, latent or toxigenic microorganisms states).
2. Fundamental transport equations: Fick, Fourier, Newton, containing *diffusion*, *thermal conductivity*, *viscosity* constants. General form: flux = constant × (driving potentials gradients).
3. Technical transport equations containing *global* coefficients: *hydraulic conductivity* or *permeation* in the filtration (Poiseuille, Darcy), *adduction* and *global thermal transport* in the thermal *convection* between walls and fluids (Newton), *mass transfer* between phases. General form: flux = coefficient × (driving potentials differences).
4. *Heat content*, *specific heat* and phase-changes *enthalpy* (melting, evaporation, sublimation etc.).
5. Classification of the possible structures in materials as related to their physical states (*solid*, *liquid*, *aeriform*) and to the *dimensional scale* of their component structure-elements. For example, distinction among the following structures at the nano-colloidal scale: "sol" (*s*/*l* dispersion), "foam" (*g*/*l* dispersion), "emulsion" (*l*/*l*'dispersion), "gel" (*l*/*s* dispersion) etc.
6. Geometrical quantities typical of the macroscopic bodies: *surfaces*, *diameters*, *lengths*, *thicknesses*, *elevations*, *heights* etc.
7. *Thermodynamic state* variables: *temperature*, *pressure* on fluids as well as *mechanical stresses* on solid materials, both normal and tangential, and *molar quantities* of the components.
8. *Non-thermodynamic* variables: microorganisms *N* concentrations (microbial *loads*), physical fields Φ *intensities* (gravitational, electric, magnetic, electromagnetic) and "space" and "time" variables functional to the space-time definition/location of the system (*cartesian coordinates* of *position*, *dimensions*, *elevations* etc. and of *age, evolution/kinetic stage*).
9. Technical equations on the determination of the fluid-dynamical *dimensionless numbers* (e.g. thermal convection Nu Nusselt number, mixing Np power number) or of the fluid-dynamical quantities of the system (e.g. mixer power).
10. "Rate equations" correspondent to the "transport equations" of the earlier ≠ 1 e 2 notes, expressed in terms of mass or heat fluxes.
11. Dynamic aspects. Kinetic equations of the Chemistry (changing systems of molecules populations), Microbiology (changing systems of microorganisms populations), Physics (changing systems of populations of excited oscillators: cooling or heating kinetics) and Physical-Chemistry (changing systems of phases and components in contact, whose molecular populations are distributed/shared between the solid and fluid states; e.g. in the course of the *l*/*s* extraction from solids with solvents). General form $Y = f(\text{constant}, t)$ where Y = populations concentrations at the time *t*.
12. Static aspects. Equilibrium equations of the specific systems of the previous ≠ 11 note, defined by simple *algebraic* equations expressed as ratios between the X, Y populations involved the course of the transformations (molecules, microorganisms, atoms/molecules distributed between phases or excited by energy starting from a stationary state) or more simply defined by single values of state variables amounts (e.g. temperature: form $T = const$ for thermal equilibrium). General form: Y/X = constant.
13. Flow critical values of the "carrier" fluids (= m/t) and heats (= Q/t) incoming or outcoming (m_{IN} inputs or m_{OUT} outputs) in those typical processes utilizing fluids: as evaporation (process steam), cooling (refrigeration liquid) or drying (removed moisture).
14. Energy yields (useful/done vs. engine/start work) or matter yields of the final out-coming products v.s the in-coming ones. General form: Y/X = (OUT quantities)/(IN quantities).
15. Components amounts to be used/mixed (m_{IN} inputs) or extracted/removed (m_{OUT} outputs) in the course of the *formulations* of products obtained from mixed "ingredients" or from depleted raw matters; solvents amounts (m_{IN} inputs) to be used in the course of the *formulations* of the chemical extractions. Matter yields in the course of the same operations; same general form of the previous ≠ 12 note.
16. *Sensible* and *latent* heat equations useful for the heat-exchanged calculations, as in the global form (net amounts in a fixed time) as in the dynamical stationary form (flows). $Q = m \times$ specific heat constant $\times$ (T difference) and $Q = m \times$ (H difference).
17. Layouts of the *Regulation and Control* circuits of the process critical variables: temperature, pressures, levels, concentrations etc. (circuit signals conveying numerical "information" as well "instructions" for "comparison of input and feedback values", valves "opening" or "locking", time steps etc.)
18. Several and differentiated *environments* within which a given technical system, as a raw matter or a final product, can get in touch in the course of its historical pathway or its overall life-cycle. Full general meaning of the context of the system: not only working processes but also use conditions or disposal surroundings, or even growing locations of its natural biological origin. System/environment interactions quantified as matter and energy exchanges governing the system transformations.

c. UML Aspects

This form of *meta*-knowledge may propose as a overlying frame to represent the specific dominion of the considered technical knowledge, aimed at implement a methodology focused on development of models as conceptual as technical (see respectively the left and right part in Fig. 1). In Fig. 2 is presented an example based on the ULM methodology, a general-purpose modeling language in the field of software engineering, applicable to varying knowledge dominions too [5].

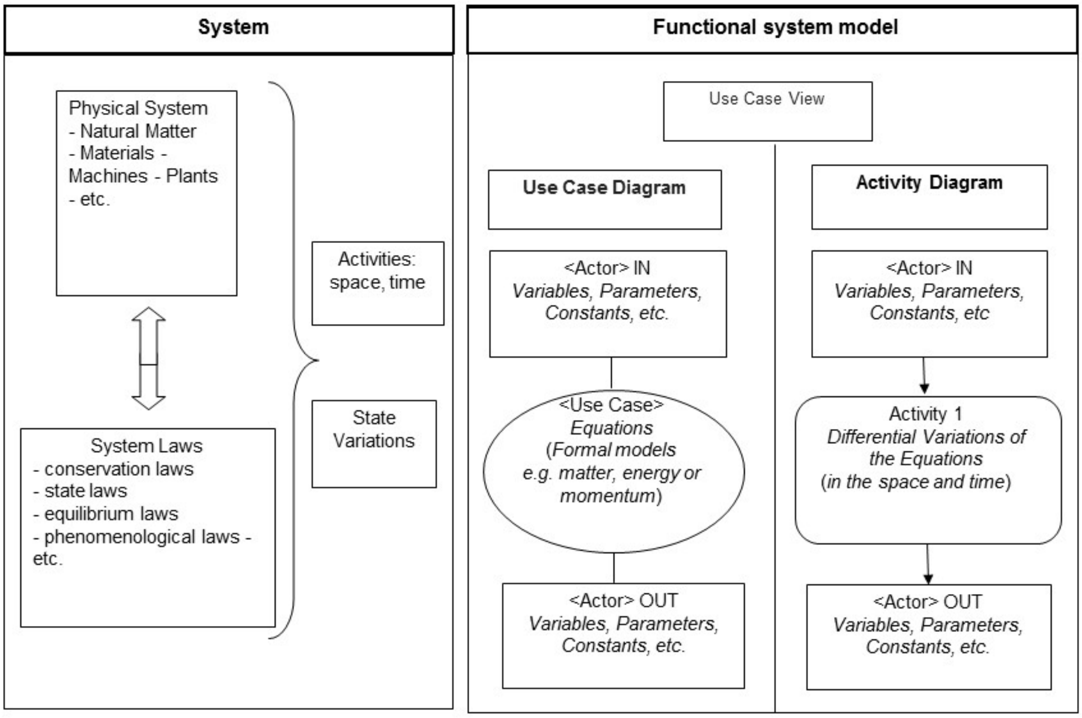


Fig. 2 - Formalization of the basic principles of the "Units Operations", explained in terms of mathematical models (equations), by means of the UML standard language. Two diagrams are clearly connected, concerning both the system (components and actions, evolutions, states) and its functional model (equations and contained parameters)

UML allows to describe three main aspects of the system concerned by the considered knowledge system:

1. its run/behavior attributes as perceived from an outside observer, as if it was though a "black-box";
2. its structure of objects/components and their reciprocal relations/connections, to the best detailed;
3. its run/behavior over time and space and the dynamics of the interactions among its objects.

Specific diagrams can be utilized in order to represent these 3 aspects, respectively: 1) Use Case Diagrams, 2) Class Diagram, 3) Sequence Diagram, Activity Diagram, State-chart Diagram.
On the left of the Fig. 2 is basically represented the system that needs to be formalized; on the right hand is instead represented an implementation by means of the standard "functional UML model", with two diagrams of the "Use Case" type. Also the more conceptual part of the system, covered on the left part of the Fig. 1, can thus be formalized by means of the ULM standard; and more is detailed the system more could be specified the diagrams of the functional mode.

Conclusion

Our work suggests that the investigated domain can be modelized by an ordinary UML class diagram, in a perspective of both knowledge acquisition and re-use; for example as an ontology map of coherent structure. I.e. utilizing in suitable way a scientific/technical network of knowledge of applicative interest.

REFERENCES

[1] D. Bell, UML basics: The class diagram. An introduction to structure diagrams in UML 2, Series in UML basics, Rational Software Architect Standard, IBM Corporation, New York, 2004.

[2] T. Ashino, Materials Ontology: An Infrastructure for Exchanging Materials Information and Knowledge, *Data Science Journal*, 2010, **9**(8), 54.
[3] M. Balaban, A. Maraee, A. Sturm, *International Journal of Information System Modeling and Design*, 2010, **4**(1),24.
[4] G. Grasso, G. Bufalo, *La Chimica e l' Industria*, 2015, **97** (6), 32.
[5] D. Bell, UML basics: An introduction to the Unified Modeling Language, Series in UML basics, Rational Software Architect Standard, IBM Corporation, New York, 2003.
[6] S. Nishadha, UML Class Diagram Relationships Explained, in https://creately.com/blog/diagrams/class-diagram-relationships/
[7] G. Grasso, *Int. J. Applied Systemic Studies*, 2017, **7**(4), 209.
[8] J. Davies, Light-weight Ontologies, in: Theory and Applications of Ontology: Computer Applications R. Poli *et al.* (Ed.s), pp. 197-229, Springer, Heidelberg - London - New York, 2010.
[9] J.D. Novak, *Instructional Science*, 1990, **19**, 29.
[10] J.D. Novak, A.J. Cañas , The Theory Underlying Concept Maps and How to Construct and Use Them, in http://cmap.ihmc.us/docs/theory-of-concept-maps.php