Jean-Daniel Dessimoz, Pierre-François Gauthey and Carl Kjeldsen, "Ontology for Cognitics, Closed-Loop Agility Constraint , and Case Study in Embedded Autonomous Systems – a Mobile Robot with Industrial-Grade Components", Proc. Conf. INDIN '06 on Industrial Informatics, IEEE, Singapore, Aug.14-17, 2006, pp6.

Ontology for Cognitics, Closed-Loop Agility Constraint, and Case Study – a Mobile Robot with Industrial-Grade Components

Jean-Daniel Dessimoz, *Member, IEEE* , Pierre-François Gauthey and Carl Kjeldsen
West Switzerland University of Applied Sciences, (HESSO.HEIG-VD)
Route de Cheseaux 1
CH-1400 Yverdon-les-Bains
Switzerland
Jean-Daniel.Dessimoz@heig-vd.ch, Pierre-Francois. Gauthey @heig-vd.ch

Abstract— **The paper refers to intelligent industrial automation. The objective is to present key elements and methods for best practice, as well as some results obtained. The first part presents an ontology for automated cognition (cognitics), where, based on information and time, the main cognitive concepts, including those of complexity, knowledge, expertise, learning, intelligence abstraction, and concretization are rigorously defined, along with corresponding metrics and specific units. Among important conclusions at this point are the fact that reality is much too complex to be approached better than through much simplified models, in very restricted contexts. Another conclusion is the necessity to be focused on goal. Extensions are made here for group behavior. The second part briefly presents a basic law governing the choice of overall control architecture: achievable performance level of control system in terms of agility, relative to process dynamics, dictates the type of approaches which is suitable, in a spectrum which ranges from simple threshold-based switching, to classical closed-loop calculus (PID, state space multivariable systems, etc.), up to « impossible » cases where additional controllers must be considered, leading to cascaded, hierarchical control structures. For complex cases such as latter ones, new tools and methodologies must be designed, as is typical in O^{3}NEIDA initiative, at least for software components. Finally, a large part of the paper presents a case study, a mobile robot, i. e. an embedded autonomous system with distributed, networked control, featuring industry-grade components, designed with the main goal of robust functionality. The case illustrates several of the concepts introduced earlier in the paper.**

I. INTRODUCTION

Automation develops, and helps controlling more and more complex systems, in industry and in economic world. People tend to think that this requires some sort of “intelligence”, something more than just bit-shifting and ALU-related operations. We feel that indeed for managing such sophisticated, complex and more abstract applications, some kind of cognition is required. The technical field of telecommunications has benefited from the precise definition of “information”, including a specific metric system, more than 50 years ago [1]. Information is still the basic “material” processed in cognitive operations today (re. Fig. 1). Over the past half-century, various attempts have been made to extend the basic approach proved successful in information domain (i. e. introducing technical definition and quantitative assessment methods) to other cognitive notions as well; re. notably [2-5]. None of those proposals however, has yet really been found appropriate or widely accepted . Nevertheless, we believe that the MCS theory (re. especially [6-8]) can be useful in that regard; the current paper aims at spreading this proposal and to make it more complete for collective notions relating to collaborative autonomous systems.

Managing sophisticated, complex and abstract applications in industry and in economic world, is not a simple task, and there are factors other than cognition to consider as well. Even though they are very much oriented towards practice, refs. [9-10] deserve to be quoted here, as they present in a short description a very inspiring, panoramic view of fundamental (feedback-)control approaches.

While the two previous domains (cognition, control) are relatively large in scope, it is also beneficial to pragmatically consider areas very close to the current state of affairs in industrial automation: O^{3}NEIDA [11-15]. O^{3}NEIDA is probably one of the best such initiatives and focuses on the incremental expansion of application domains characterized until now by basic, mostly Boolean, PLC-level operations. Although the potential advantages are excellent, it is worth discussing here also the limits inherent to this approach, and ways to go beyond them.

Our group has gained some expertise in the design of autonomous and cooperating, mobile robots. The overlap with O^{3}NEIDA domain is obvious [e.g. 16-18], as they feature embedded systems with distributed, networked control and high potential economic significance. Moreover they are still reasonable in complexity (much simpler for example than the federated enterprise of ref. [13]) and therefore provide a good test-bed for practicing concretely the quantitative assessment of selected cognitive entities.

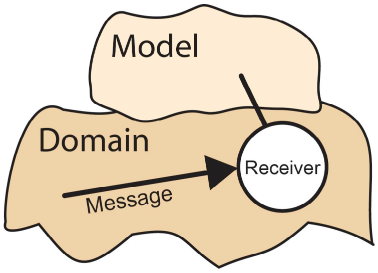

Fig. 1. Essential ingredient for a cognitive agent. information (messages) allows a receiver to build up or update its model of some domain

The paper organization reflects the same structure as this introduction: Part II presents the basics of an ontology for automated cognition, with new contributions relating to collective concepts. Part III reminds the broad perspective of feedback control, which helps understand the overall structure of complex automated systems. Part IV discusses this issue in O3NEIDA context, and Part V details a study case, centered on autonomous mobile robots.

II. ONTOLOGY FOR AUTOMATED COGNITION

An ontology for automated cognition (cognitics) is summarized in the Appendix A. Based on information and time, cognitive concepts such as complexity, knowledge, expertise, learning, intelligence abstraction, and concretization can be unambiguously defined, along with corresponding metrics and specific units.

The main difficulties lie within the classical, information-centered notions, and are just inherited in MCS cognitive definitions: information quantities in a message depend on time (history), and on specific receivers (perceived probabilities). Reality cannot be directly addressed; only models can.

Among important conclusions is the fact that reality is much too complex to be approached better than through much simplified models, in very restricted, appropriate contexts. Another conclusion is the necessity to be focused on goal.

As complexity grows, agents tend to team up thus yielding groups. In this case new notions appear (spirit and culture). Communication between members becomes a necessity (re. fig. 2).

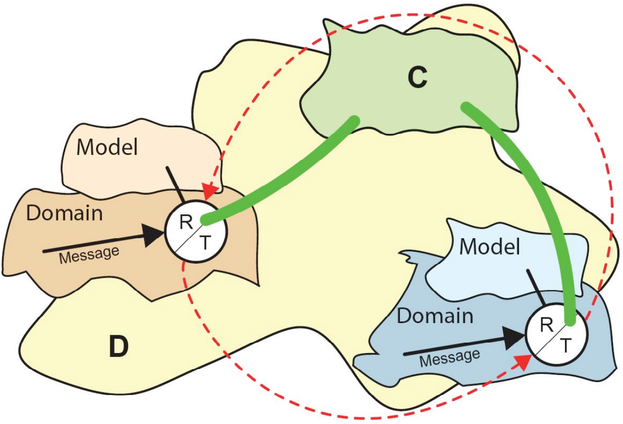

Fig. 2. In a group, agents must communicate with each other (R: receiver; T: transmitter), share a culture (C), common system of shared references, values and objectives, in reference to some common domain of interest (D) . At a meta-level, the individual members may be considered as merging, to yield a new individual (the group) with its own collective model (C).

As an example, consider an orchestra playing without conductor: the group is the orchestra, members are musicians, and the "spirit" is what makes it possible for the musicians to play together in a coherent way, even when there is no additional conductor nor outside regulating factor.

A group is a collective cognitive agent rather than an individual one. Definitions and metric equations in the MCS theory apply equally to individuals and to groups. The behavioral model adopted in MCS can be applied at the global level of an entity, but also in sub-systems, or on the contrary at a higher level of a collectivity. The group can be characterized dynamically by its "spirit" and statically by its "culture" which specifically bind together group members. Spirit and culture can be viewed as some set of intangible underlying factors that ensures the coordination of individuals so as to achieve a specific collective identity and behavior. "Spirit" and "culture" consist in a system of common, shared references, values and objectives, which may dynamically evolve, and yet possibly do not exist per se, i.e. out of the members.

III. CONTROL CLASSES

Depending on circumstances, a system may be more or less easy to control. Open-loop control, when applicable, may be the most favorable case. Often, closed-loop control provides the adequate solution. Here however, computation time and communication delays play a critical role. As Fig. 3 shows, whereas for relatively "slow" systems to control, Boolean, on-off (relay) action is adequate, for relatively fast systems, no solution exist. In the latter case, at least part of the control must be "subcontracted" to faster, agile resources, working at their own pace, autonomously.

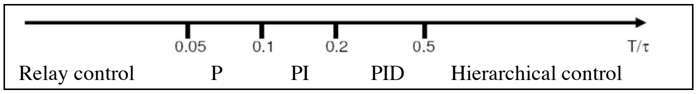

Fig. 3. Depending on the ratio of control time T (including delays in the loop) versus time constant τ of a system to be controlled, .the case may range from most easy to handle (left part), to impossible to handle with a single controller (right part). In between, for a decade interval, PID controllers may be helpful. (credit for part of Fig. : [9])

IV. O^{3}NEIDA PERSPECTIVE

Automation implies more and more specialized software and ICT's. For this reason the O^{3}NEIDA model is interesting. At all scales, from elementary devices to global enterprises it shows the important role of software components, of protecting the associated IPR's, and of reusing existing solutions.

Refering to Section II, the O^{3}NEIDA model provides a precious elements of culture for all the actors of value-creating networks involving automation, including for example access to detailed, shared world description. Not only the framework is useful in conceptual and knowledge terms, but moreover it often allows to operationalize this knowledge with appropriate standards in representation and execution platforms. Furthermore it binds incremental contributions with elementary property rights, thus paving the way for fair economical integration and cooperation.

Nevertheless aspects other than software components also require attention (re. Fig. 4). Not all can be encapsulated in software; for example it is critical to bridge the physical world and soft representations, and therefore elements such as sensors, actuators, human interfaces cannot be neglected. Even when considering ICTs, many strong constraints exist which in practice inevitably breaks software into many fragmented domains. As an analogy, consider the necessary tradeoff between code size and addressing range. More generally, Fig. 3 underlines the necessity to adapt control agility to application-specific dynamic properties.

As will be illustrated below, application-specific constraints may for example force the design and use of a new FPGA instead of relying on an existing C++ component to be executed on a computer. Or they may require the implementation of control loops in dedicated hardware and DSP, instead of running 1131 code on a regular PLC .

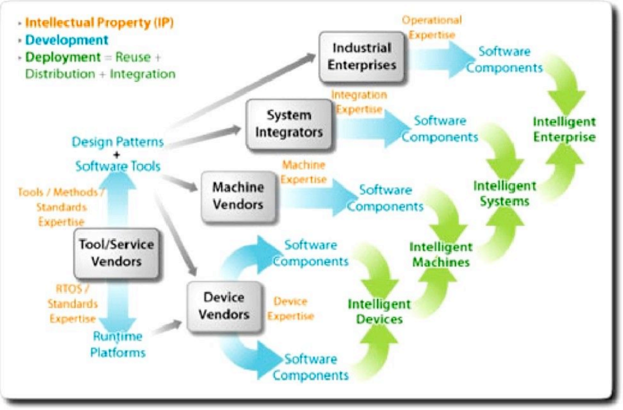

Fig. 4. O^3NEIDA provides a good model for taking into account software components and tracking their value across production systems and products ([14]). Other aspects also require attention such as hardware, architecture, groupware , methods and techniques, as well as related IPR and repositories.

V. CASE OF MOBILE ROBOTS

The case of mobile robots illustrates here several of the concepts introduced in the first parts of the paper, cognitics and closed-loop control architecture. We refer to the ARY class of robots, developed in our lab at Yverdon-les-Bains. Even though they may be quite unique in some aspects (e.g. proprietary Piaget environment and extremely fine-grained multi agent time sharing), they are also quite representative of many other current mid-size autonomous and/or cooperating robots.

ARY robots are mobile, autonomous systems, with multiple embedded components, mostly made of industry-grade components, designed with the main goal of robust functionality (re. fig. 5). In O^3NEIDA perspective, the point of view adopted here is closest to the one of a system integrator, but may also be illustrative of a machine vendor or of an industrial enterprise point of view.

Unfortunately, some faster processes require their own local control resources, and the heterogeneity of tasks to be handled prevent the system designer to rely on purely object-oriented solutions, in the usual computer engineering sense.

Due to the complexity of the application, the variety of functions to be implemented, each with their own dynamic specifics and IP management requirements, a distributed, multiprocessor, networked control structure had to be designed (re. fig. 6).

At the supervising level an application-oriented programming environment, with multiple agents, had to be originally designed, Piaget. It now runs on MS Windows XP Pro operating system. The fundamental advantage of this type of approach is the ease and speed of changes in task specifications to be done by users, which must be done in order to reflect changes in task configuration or in solving strategies. Let's be quantitative as advocated in Section 2: The fundamental limit on reusability is set by the huge number and integral complexity of possible applications (e. g. in Eurobot framework, each year a new challenge is set, which each time requires Mb's of specification. In practice, the chance of overlap in cognitive domains over the years tends towards zero. Each year new application specific instructions must be implemented, such as "ChooseBridgeVisually" one year, or "ShootAtTower(GGG)" another year; similarly, unique application-specific subsystems and constructions must be elaborated, featuring sensors, actuators as well as structural elements).

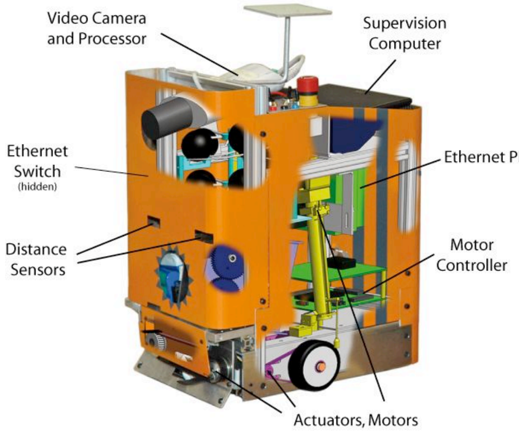

Fig. 5. Dude, one of ARY robots, with representation of major hardware components. At highest level, distributed processors communicate with each other mainly with Ethernet TCP/IP protocol. At lower levels, respective functions are quite autonomously implemented.

The highest abstraction elements (supervision computer, camera, motion control unit and I/O processor) communicate with each other in Ethernet/TCP-IP mode, via hub or switch, sometimes conveying fieldbus (Modbus) protocol. When time and volume constraints allow, this type of solution is best, especially in terms of availability of components, reusability, tracking of IPR for standard elements. Sometimes however, requirements are more stringent in terms of time and/or volume. Consequently, control must be distributed and ad hoc subsystems designed. For example our Alf robot had about 10 small detachable parts (<<10 cm side size), most of them autonomously able to move and perform adaptive tasks based on real-time sensing; communication was ad hoc, optical.

At medium level, a PLC is generally found attractive, Not only PLC components are designed for robustness (electric, magnetic, mechanical, chemical, logic…), compactness, modularity but also they can be distributed, work autonomously and prove very reliable in time regularity for the 1ms-0.1s range. The weakest points today maybe, depending on applications, size, speed, power, connectivity, protocol-compatibility; but again, in view of the infinite number of possible applications, an infinite number of limits could also be stated …

At a lower level, mobile robots generally require motor control loops, with feedback of various nature such as current, speed or position. Here the typical time range goes from 10 microsecond to 1 ms. Usually total tracking ability (re. odometry) and stability are required, and therefore specialized controllers must be implemented. The mix of short sampling time for some aspects and relatively complex tasks for motor management, motion laws and trajectory control (multi-axis kinematic synchronization) make this area hard to standardize, either in hardware, architecture or software aspects

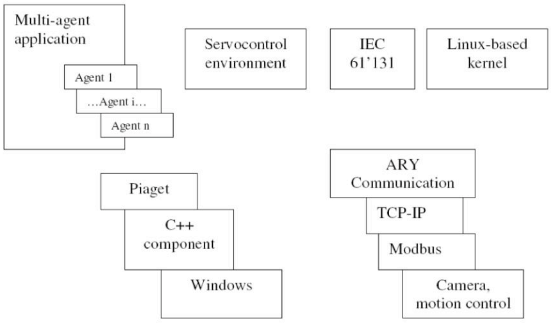

Fig. 6. ARY software components are very much diverse. On the left the structure on supervisory computer; on the right, modules fit local conditions in distributed processors.

At the very lowest level, some components are integrated, with hardware or firmware solutions, with an excellent reusability, being found on the market, and an excellent IP protection as well, expertise being encapsulated in non-documented micro-electronic components and transducers. From an integrator perspective however, there is no software component here, and repository is in practice rather a spare part container or retailer’s stock than a soft database.

Let’s assess quantitatively some of the cognitive capabilities of ARY robots. Abstraction is the ratio of input information quantity to output information quantity. For example the detection of robot location (x, y, α, with 1%o accuracy, i.e. 30 bit) on the basis of an acquired image (320 columns by 240 lines, with RGB encoded pixels, each component 8 bit) is a process featuring an abstraction index of about 61’440. Expertise is the ability for a cognitive system to take decisions fast and right. Relevant units are lin/s. It is the product of knowledge, K and fluency, f. Considering the specific example above for position estimation, K=log$_2$(30*2**1’895’040)) , i.e. about 2 million lin. Considering that image acquisition and vision processing takes about 0.1 s, f has a value of 10. Therefore the expertise quantity of ARY in this domain is of about 2*10**7 lin/s.

VI. CONCLUSION

The paper is at the crossing point of several avenues.

1. One avenue relates to cognitive domain, with firm definitions and metric system allowing to compare alternative solutions, estimate progress and assess in absolute terms the performance levels of automated (or not) cognition resources, under their essential aspects: knowledge, expertise[1], learning, abstraction, complexity, etc. As well as, reciprocally, to assess task requirements in cognitive domain. Current progress allows to consider collective agents, groups, cooperating in cognitive domains.

2. Another avenue is control, and particularly feedback control, a very important component of automation. Theories show that the relative agility of a control system may yield solutions ranging from effective and simple (threshold–based relay technique) to totally impossible (“subcontracting” at least some part of the task to an autonomous, more agile control subsystem is then required), depending on dynamic properties of the system to be driven

3. The O^{3}NEIDA initiative is very precious in setting-up the framework of a culture for an important part of collective cognitive systems in automation and control. Software aspects, knowledge reuse, value tracking, and even some hardware considerations for standard support platforms may be rather well addressed in O^{3}NEIDA context. At the moment however other, complementary reference models, are still required for domains where agility, systemic constraints, historical bias (or maybe some IPR protection strategies) call for resource classes other than software in currently proposed standard (e.g. hardware, groupware, ad hoc techniques and methodologies).

4. ARY robots provide an interesting field of experimentation for the identification of major problems (relevant research directions) and of effective solutions (validation), in automated cognition. They consist in autonomous and/or cooperating systems, with embedded cognitive resources, subject to numerous physical and real-time constraints, representative of most automation and control issues in industry and service applications. Several time and size scales need to be simultaneously considered in a coherent way (holistic or systemic approach), calling each for very different types of solutions. Fast, distributed hardware components and transducers complement middle-scale resources (PLC, computerized camera, specialized servo-controller, FPGA), which in turn support “slower” (less reactive) but more general and integrative ICT elements (Ethernet, TCP/IP, Windows XP-Pro, compact PC, Piaget environment and multi-agent application) .

The authors hope that the current contribution will prove

---

[1] Expertise is probably the most crucial cognitive quality, and a “B-prize” may well be defined one day in connection with this topic.

useful especially in motivating many potential beneficiaries to make use of MCS model, and allowing them to concretely get started within given context and examples. Further information can be provided upon request by the authors.

Furthermore, cognition related quantitative attributes could be a very significant components of O^3NEIDA repositories and standards.

APPENDIX A - SUMMARY OF MCS CORE DEFINITIONS

The appendix briefly presents, by alphabetical order, the core definitions in MCS - our model for cognitive sciences. The number behind each concept denotes the logical order in which definitions are introduced (re. above the novel definitions).

Abstraction (3b). Property of a system that generates less information than it receives. The abstraction index, i_{abs}, is the ratio of incoming information quantity (n_i [bit]). over the out coming information quantity (n_o [bit]). Inverse of concretization. Equ.: $i_{abs}=n_i/n_o$ [without unit]

Bad (8f) Bas is the contrary of "good" (re. art. good).

Complexity (3a). Property of a model that requires a lot of information in order to be exhaustively described. Quantitatively, complexity is the amount of required information. Unit: same as for information, i.e. [bit]

Concretization (3c): Property of a system that generates more information than it receives. The concretization index, i_c, is the ratio of out coming information quantity (n_o [bit]) over the incoming information quantity (n_i [bit]). Inverse of abstraction. Equ.: $i_c=n_o/n_i$ [without unit]

Culture (12a): re. to art. group.

Experience: (4b) Property of a system that has been exposed to a cognitive domain. Quantitatively, it is usually evaluated in terms of time (duration) [s]. An alternate (better?) view is to assess experience, R, in terms of number N_a of witnessed input-output associations. Equ.: $R=N_a*(n_i+n_o)$ [bit]

Expertise (5a). Property of a cognitive system which delivers fast the pertinent output. Quantitatively, it is the product of knowledge, K, and fluency, f. Equ.: $E=K*f$. The unit is [lin/s]. In general terms, synonyms for expertise include know-how, skill, competence and excellence.

False(8e): Contrary of "true" (re. art. true)

Fluency(4c): Property of a system which delivers information fast. It can be viewed as a processing speed. Fluency, f, is the inverse of the time duration , Δt, necessary to deliver output information. Equ. : $f=1/\Delta t$ [1/s]

Good (8c): Good is defined on the basis of "right" (re. art. right): "Good" is "right" when the law to comply with is "to progress towards a chosen goal". For example, if a robot is required to move, it is good for it to switch on some power circuits.

Group (11). A group is a collective cognitive agent rather than an individual one. Definitions and metric equations in the MCS theory apply equally to individuals and to groups. The behavioral model adopted in MCS can be applied in sub-systems, at the global level of an entity, as well as at a higher level of a collectivity. The group can be characterized dynamically by its "spirit" and statically by its "culture" which specifically bind together group members. Spirit and culture can be viewed as some set of intangible underlying factors which ensure the coordination of individuals so as to achieve a specific collective identity and behavior. "Spirit" and "culture" consist in a system of common, shared references, values and objectives, which may dynamically evolve, and yet do not exist per se, i.e. out of the members.

Information (2). Information is what allows a receiver to update his model. Quantitatively, it is the difference of model size in terms of information content, between the states "before" and "after" message arrival. Computation is made on the basis of message probabilities, which are essential elements in the model Consider that the incoming message is one among N possible variants. If the probabilities of those various occurrences of the message are pi, where pi is the probability of the ith message, then the average quantity Q_a is given by the following equation: $Q_a = \sum_{i=1}^{N} p_i \log_2\left(1/p_i \right)$ The log is usually taken in base 2, thereby yielding [bit].

Intelligence (7). Intelligence is the property of a system capable of learning. In quantitative terms, intelligence can be assessed as an index, i, which is the ratio of learning with respect to experience. Depending on the intuitive or more rigorous choice of formulations introduced for experience, we have two equations. Equ.: $i=L/\Delta t$ [lin/s^2] (or $i=L/\Delta R$ [lin/s/bit])

Knowledge (4a). Knowledge is the property of a system which delivers the pertinent output, either proactively or in response to incoming messages. Quantitatively it is given by the following equation: $K=\log(no*2^{ni}+1)$. The log is in base 2, and the unit is the [lin].

Learning (6). Learning is the property of a system capable of increasing its expertise level as time goes (or better: as experience goes). Equ: $L=E(t1)-E(t0)$. Alternate view: $L=E(r1)-E(r0)$. In both cases the unit is [lin/s]

Model (1). In general terms, a model is a simplified (that is, incomplete by essence) representation of reality, which is found useful in order to reach some specific goal. In MCS the basic reference model is behavioral. It can be viewed as a kind of (virtual) table, which contains as many states as possible incoming message types; each state contains the instant probability of occurrence for the corresponding input message, and also contains the corresponding output message. The goal of this model is to allow the quantitative assessment of key cognitive properties, such as knowledge, expertise, or learning.

Reductibility (5b): Property of a system which can be implemented by subsystems of integral complexity smaller than the complexity of the system itself .

Right (8a) "Right" is usually considered as a logic, Boolean value, complementary to "wrong". Let us define "right" as the qualifier of an entity that complies with a given law (assertion). For example if the law is "to move ahead", a step forward is "right".

Sapience (10) is the essential property of a cognitive agent, i.e. of an active structure capable of cognition. It appears under a number of signs, such as knowledge, expertise, or intelligence (already defined and made measurable in MCS). Quantitatively, sapience may be characterized by an index, in reference to humans ("homo sapiens"). Sapience (index) is thus a ratio; no specific unit.

Simplicity (4d): Property of a model which requires little information in order to be exhaustively described. Quantitatively, simplicity, exactly like complexity is the amount of required information. Unit: inverse of information unit, [1/bit].

Spirit (12b): refer to art. Group.

True (8b) can be defined on the basis of "right" (re. art. right): "True" is "right", when the law to comply with is "correspondence to reality". For example it is true that braking reduces speed.

Wisdom (9) is a specific property of cognitive agents, which refers to their ability to take good decisions, i.e. to be expert in delivering the messages that make an agent reach a given goal. To make it simple and easy, we propose here to estimate in Boolean terms the quantity of wisdom for an agent, on a given domain: true or false, reflecting the fact that the goal is being reached or not by the agent. Without being essential, a usual feature of wisdom is to relate to complex situations and major or "meta"-goals: to survive, to win the game, to gain a place in the Hall of Fame.

Wrong (8d) is the contrary of right (re. art. right).

REFERENCES

[1] Claude Elwood Shannon, *coll. papers*, ed. by N.J.A. Sloane, A. D. Wyner, Piscataway, NJ, IEEE Press, 1993

[2] G.J.Chaitin,Scientific American 232(1975)47-52.

[3] A. N. Kolmogorov, Information theory and the theory of algorithms , ed. by A. N. Shiryayev, transl. from the Russian by A. B. Sossinsky Dordrecht (etc.) (Kluwer Academic Publishers, 1993)

[4] D.Michie, *On Machine Intelligence*, 2nd Ed., (Ellis Horwood Ltd, Chichester, W Sussex, England, 1986).

[5] S. J. Rosenschein, Formal Theories of Knowledge in AI and Robotics, New Generation Computing, Ohmsha Ltd and Springer Verlag, 3 (4)(1985)345-357.

[6] J.-D. Dessimoz, "Knowledge in formulas", SGAICO Newsletter, *SI Information*, Soc. Suisse des Informaticiens, Zürich, Aug. 1991.

[7] J.-D. Dessimoz and Giovanni Mele, "Performance assessment of cognitive systems; case of elementary mobile robots", *Proc. ECAI 94, 11th Europ. Conf. on Artif. Intelligence*, Amsterdam, 7-12 Aug., A. Cohn. ed., John Wiley & Sons, New York, pp. 689-693, 1994.

[8] Jean-Daniel Dessimoz, "About the Necessary Move from Cognitics to Ethics; Additional Definitions, and Contributions to Metrics in MCS", *DARH-2005 – 1st Intern. Conf. on Dextrous Autonomous Robots and Humanoids*, with sponsorship Eurobot, IEEE, CLAWAR and CKI, HESSO.HEIG-VD (West Switzerland University of Applied Sciences, Yverdon-les-Bains, Switzerland, Nay 19-22, 2005,

[9] Sermondade, C. et A. Toussaint,"*Régulation , tome I* ", Etapes, Nathan ,pp.128,1994

[10] Sermondade, C. et A. Toussaint,"*Régulation , tome II*", Etapes, Nathan, pp.128,1994,

[11] R.W. Lewis, *Modelling control systems using IEC 61499*, SBN: 0852967969 - IEE, Control Engineering Series, U.K, 2001, pp 208.

[12] Bernardo Ferroni, « IPLnet: Internationale Aktivitäten und O3NEIDA Dokumente », Icimsi-SUPSI , Lugano, Switzerland, private communication, 2 février 2005

[13] George Brown, IT Research, Intel "Collaborative Decision-Making in a Federated Enterprise", *Knowledge Exchange Partnership (KEP) Digital Innovation Forum*, December 2, 2004, pp. 24

[14] Vyatkin, V., Christensen, J., Lastra, J.L. 'An Open, Object-Oriented Knowledge Economy for Intelligent Distributed Automation', *IEEE Transactions on Industrial Informatics, 1*, (1), p4-17, 2005

[15] Allan Martel, IMS Canada, "Manufacturing 2020 in Canada ", incl. O3NEIDA, *Conf. on Advanced Manufacturing*, IMS NoE, Leuven,Belgium,16-17nov. 2005

[16] A. Trad, J.-D. Dessimoz. « Proactive Monitoring and Maintenance of Intelligent Control SystemsApplication for Mobile Robot Systems », *Proc. 2nd IEEE Intern. Conf. on Industrial Informatics INDIN'04*, Collaborative Automation One Key for intelligent industrial environment, Fraunhofer Inst. Produktionsanlagen und Konstruktionstechnik,24-26 June 2004, Berlin, Germany

[17] Nicolas Uebelhart, Florian Glardon and Pierre-François Gauthey, "Lomu, an Autonomous Mobile Robot with Robust Architecture and Components", *« DARH-2005 - 1st International Conference on Dextrous Autonomous Robots and Humanoids»*, with sponsorship Eurobot, IEEE, CLAWAR, and CTI, HESSO-HEIG (West Switzerland University of Applied Sciences), Yverdon-les-Bains, Switzerland, May 19-22, 2005. (re http://www.darh2005.org")

[18] André Perrenoud, Pierre-François.Gauthey, Nicolas Uebelhart, Jean-Daniel Dessimoz, "Development and Opportunities for Mobile Robots in Switzerland", *IPLnet 2005: "Needs and Opportunities for Swiss Industry"*, 5th national Workshop of the Swiss Network of Excellence for Integral Production automation and Logistics, IPLnet, Schloss Böttstein, Sept. 5-7, 2005

[19] Description of RobocupAtHome Goals and Modalities: available on: http://www.robocupathome.org (2006).