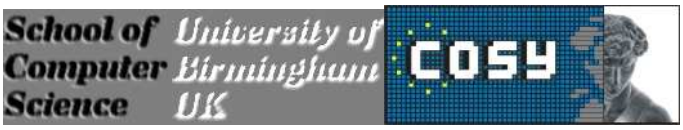

A First Draft Analysis of Some Meta-Requirements for Cognitive Systems in Robots

(An exercise in logical topography analysis.)

Aaron Sloman and David Vernon

Last updated: 16 Nov 2008; 25 Apr 2010; 15 Jul 2013; 25 Jul 2013
This file can be referenced at
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/meta-requirements.html
A messy automatically generated PDF version is here.

This is also referenced on the Birmingham Cosy Project web site:
http://www.cs.bham.ac.uk/research/projects/cosy/papers/#dp0701

NOTE ON FORMATTING:
Adjust the width of your browser window to make the lines the length you prefer.
This web site does not attempt to impose restrictions on line length or font size.

Contents

- Good and bad varieties of persistence
- Other topics to be added
  - Predictability/unpredictability
  - Stability (suggested by Bernhard Sendhoff -- Honda Research)
  - Variability (suggested by Bernhard Sendhoff -- Honda Research)

    The next lot require meta-semantic competences.
  - Prudence/discretion
  - Considerateness/Kindness/Thoughtfulness
  - Forgivingness/Vengefulness
  - Honesty
  - Adventurousness/Risk-aversion
  - Impulsive/Restrained
  - Courage/Boldness vs timidity
  - Playfulness
  - Morality?
  - .....

Introduction and background

This is a contribution to construction of a research roadmap for future cognitive systems, including intelligent robots, in the context of the euCognition network, and UKCRC Grand Challenge 5: Architecture of Brain and Mind.

A meeting on the euCognition roadmap project was held at Munich Airport on 11th Jan 2007. Details of the meeting, including links to the presentations are available online at http://www.eucognition.org/six_monthly_meeting_2.htm.

It is often assumed that research starts from a set of requirements, and tries to find out how they can be satisfied. For long term ambitious scientific and engineering projects that view is mistaken. The task of coming up with a set of requirements that is sufficiently detailed to provide a basis for developing milestones and evaluation criteria is itself a hard research problem. This is so both in the context of (a) trying to produce systems to elucidate scientific questions about intelligent animals and machines, as in the UKCRC Grand Challenge 5 project or (b) trying to advance long term engineering objectives through advancing science, as in the EU's Framework 7 Challenge 2: "Cognitive Systems, Interaction, Robotics" presented by Colette Maloney here.

> An elaboration of the EU FP 7 challenge is now available here:
> ftp://ftp.cordis.europa.eu/pub/ist/docs/cognition/fp7-challenge2-background_en.pdf

An explanation of why specifying requirements is a hard problem, and why it needs to be done, along with some suggestions for making progress, can be found in this presentation:

http://www.cs.bham.ac.uk/research/projects/cosy/papers/#pr0701
"What's a Research Roadmap For? Why do we need one? How can we produce one?"

Working on that presentation made me realise that certain deceptively familiar words and phrases frequently used in this context (e.g. "robust". "flexible", "autonomous") *appear* not to need explanation because everyone understands them, whereas in fact they have obscure semantics that needs to be elucidated. Only then can we understand what the implications are for research targets. In particular, they need explanation and analysis if they are to be used to specify requirements and research goals, especially for publicly funded projects.

First draft analyses are presented below. In the long term we would like to expand and clarify those analyses, and to provide many different examples to illustrate the points made. This will probably have to be a collaborative research activity.

After a draft of this paper was written David Vernon drew my attention to the Software Engineering research literature on 'ilities' mentioned in a note below.

The 'ilities' (pronounced 'ill'+'it'+'ease)'are things like 'flexibility', 'usability', 'extendability'. It seems that software engineers have been discussing them for some time and regard them as expressing 'non-functional' specifications. In contrast, we suggest they are higher-order (or schematic) functional specifications, as explained in this document.

Towards a generic analysis

Many of the words discussed below refer to what could be called "meta-requirements" (or possibly "schematic requirements"). What that means is that none of the labels is directly associated with a set of criteria for meeting the implied requirements. For instance, the meaning of 'robust' does not specify features (whether physical properties or behaviours) that would justify the application of the label. Rather the meaning of such a word is more abstract.

Instead of expressing a concept that specifies criteria for instances, a word like 'robust' or 'flexible' expresses a concept that specifies *ways of deriving* criteria or requirements *when given a set of goals or functions.* (Such a concept could be called a "meta-concept", a "higher-order concept" or a "schematic-concept".)

There are many words of ordinary language that are like that, as philosophers and linguists have noted. For example, if something is described as "big" you have no idea what size it is, whether you would be able to carry it, kick it, put it in your pocket, or even whether it is a physical object (for it could be a a bit idea, big mistake, or a big price reduction). If it is described as a big pea, a big flea, a big dalmation, or a big tractor, etc. you get much more information about how big it is, though the information remains imprecise and context-sensitive.

In that sense "big", when used in a requirement, specifies a meta-requirement: in order to determine what things do or do not satisfy the requirement, a meta-requirement M has to be applied to some other concept C, and often also W, a state of the world, so that the combination M(C, W) determines the criteria for being an instance in that state of the world. Without W, you get another meta-requirement or schematic requirement M(C) that still requires application to a state of the world to produce precise criteria. E.g. what counts as a big tree, or a big flea, can depend on the actual distribution of sizes of trees or fleas, in the environment in question.

Thus the combinations Big(Flea) and Big(Tree) determine different ranges of sizes; and further empirical facts (about the world) determine what counts as a normal size or a larger than normal size, in that particular state of the world, or geographical location. In another place the average size of fleas or trees might be much larger or much smaller.

It's more subtle than that, because sometimes in addition to C, the concept, and W, the state of the world, a goal or purpose G must also be specified, in order to determine what counts as "big enough" (e.g. a big rock in a certain context might be one that's big enough to stand on in order to see over a wall, independently of the range of sizes of rocks in the vicinity). Often we don't explicitly specify W or G, because most people can infer them from the context, and use what they have inferred to derive the criteria. Notice that such meta-requirements can be transformed in various ways, e.g. 'big enough', 'very big', 'not too big', 'bigger than that', etc. using syntactic constructs that modify requirements.

> The point that communication often uses words and phrases whose meaning has to be combined with non-linguistic information available to either or both of speaker and hearer is elaborated in more detail in this draft discussion paper
> Spatial prepositions as higher order functions: And implications of Grice's theory for evolution of language.

Another example is 'efficient'. If you are told that something is efficient, you have no idea what it will look like, feel like, smell like, what it does, how it does it, etc. If it is described as an efficient *lawnmower*, or an efficient *supermarket check-out clerk*, or an efficient *procedure for finding mathematical proofs*, then that combination of meta-concept 'efficient' with a functional concept (e.g. 'lawnmower') will provide information about a set of tasks or a type of function, and what kinds of resources (e.g. time, energy, fuel, space, human effort, customer time) the thing uses in achieving those tasks or performing those functions. Someone who understands the word 'efficient', knows how to derive ways of testing whether X is efficient in relation to certain tasks or functions, by checking whether X achieves the tasks or functions in question well, while using (relatively) few of the resources required for the achievement. The object in question will not in itself determine the criteria for efficiency: the object in your shed may be an efficient lawnmower but not an efficient harvester. Or it could be an efficient doorstop in strong winds while being an inefficient lawnmower.

> The word 'relatively' was added in parentheses because whether something is efficient sometimes depends on what the competition is -- just as whether something is big sometimes depends on what the competition is. Something that is efficient at one time may turn out to be highly inefficient later because much better versions have been developed. This is related to the meaning of "better", analysed in *How to derive 'better' from 'is'* (1969).

There are many meta-concepts in ordinary language which are often not recognised as such, leading to pointless disputes about their meaning. But this paper deals only with a small subset relevant to requirements for intelligent machines.

Meta-Requirements

My claim is that many of the words used apparently to specify requirements are actually labels for meta-requirements in the sense explained in the previous section. That means that they have to be combined with further information in order to generate actual requirements. For instance, a meta-requirement M, may have to be applied to a concept C, defining some class of entities with a function or purpose, a state of the world W, and possibly also a goal G, for which instances of C are being chosen. Then the combination M(C,W,G) can determine a set of criteria for satisfying the meta-requirement.

For each meta-requirement M, the criteria will be determined in a specific way that depends on M. So different meta-requirements, such as ’robustness’, ’flexibility’, and ’efficiency’, will determine specific criteria in different ways. Each one does so in a characteristic uniform way, just as "efficient" has roughly the same meaning (or meta-meaning) whether combined with "lawnmower", "proof procedure" or "airliner", even though in each case the tests for efficiency are different. Likewise "big" has the same meta-meaning when applied to "flea", "pea", "tree" and "sea", even though the size ranges are very different. (Though its use in connection with "idea" or "mistake" is more complex.)

It is a non-trivial task to specify the common meaning, or the common meta-requirement, for a word referring to meta-criteria for cognitive systems. So what follows is an incomplete first draft, which is liable to be extended and revised. This draft will need to be followed up later with detailed examples for each meta-requirement. We start by giving a list of commonly mentioned meta-requirements and provide a first draft ’high level’ analysis for each of them.

Each of the meta-requirements is capable of being applied to some category of behaving system. This system may or may not have a function, or intended purpose, though in most cases there is one or a set of functions or purposes, assumed by whoever applies the label naming the meta-requirement. For example, if we talk about a domestic robot that deals flexibly with situations that arise, then we are presupposing a specific (though possibly quite general) function that the robot is intended to serve in those situations. So the meta-requirement, in combination with the function allows us to derive specific requirements concerned with forms of behaviour, or more generally with kinds of competences that are *capable* of being manifested in behaviour, even if they are not actually manifested. I.e. the derived criteria are *dispositional*, not *categorical*.

This is closely related to the notions of "polymorphism" and "parametric-polymorphism" used in connection with object-oriented programming, where there are different classes of objects and certain functions, predicates or relations are capable of being applied to one or more class-instances at a time, with results that depend on the types of the instances (the parameters).

For example, the concept "X gave Y to Z" allows the variables, X, Y and Z to be instantiated by different sorts of entity: e.g. X and Z can be humans, other animals, families, communities, corporations, nations, and what makes an instance of the schema true can depend in complex ways what types of entity X, Y and Z are. Consider what happens when X is not a donor in the normal sense, but something abstract (X) that gives an idea (Y) to a thinker or reader (Z).

It should be obvious that many of the meta-requirements below exhibit such parametric polymorphism, e.g. "X is safe for Y to use", "X can easily teach the use of Y to Z", where X is some abstract tool.

This concept of polymorphism is relevant to many meta-requirements.

Meta-requirements and behaviour envelopes

All the meta-requirements for future robots discussed here assume that the functions of the robots define a collection of possible behaviours (or task+behaviour pairs). That collection will have have an "envelope", (or "bounding envelope") with the relevant behaviours within the envelope and other possible behaviours, that are either beyond the machine’s capabilities, or are never relevant to the goals or functions, are outside the envelope. We can use that idea as a framework for a first draft specification of at least a significant subset of meta-criteria.

The generic formula is:

> *Given such an envelope E for a set of behaviours, and a meta-criterion M, the application of M to E, M(E) produces some result which is a modified specification for the set of behaviours -- e.g. expanding or contracting the set of behaviours, or the set of transitions between behaviours.*

In other words, our meta-criteria are concerned with features of what the machine can do in relation to the envelope: e.g. how varied the behaviour transitions are within the envelope, and how the machine can extend or modify the envelope over time. E.g. a meta-requirement transforms the specified behaviour envelope in a systematic way to produce a new set of functional requirements. What that means will differ according to what the set of behaviours and purposes is, and what the meta-criterion is.

The concrete requirements derived from the meta-requirements all relate to a space of circumstances in which behaviour can occur and a space of possible behaviours in those circumstances. Given a specification of the behaviour envelope, the notions of robustness, flexibility, etc. determine requirements for the behaviours and the envelope, but they do so in different ways.

Some of the meta-requirements are concerned only with

1. *what happens within the envelope*

whereas others are concerned with

2. *possible changes to the envelope,*

   or

3. *the system's ability to make changes to the envelope without being 'led' to make them by an external influence such as a teacher.*

4. *The speed or other features of the changes to the envelope.*

This will now be made clearer by showing how the meta-requirements differ in their implications.

Disclaimer regarding the analyses presented

No claim is made here that the analyses below provide definitions of the words as they are ordinarily used.

This is not a lexicographical exercise to determine what should go into a dictionary (though dictionary makers are welcome to make use of this). Rather it is an exercise in what has been labelled the study of 'Logical topography', which is a modified version of Gilbert Ryle's notion of 'Logical Geography'. The difference is explained in this Web document: Two Notions Contrasted: 'Logical Geography' and 'Logical Topography' Variations on a theme by Gilbert Ryle: The logical topography of 'Logical Geography'.

Roughly, 'logical topography' refers to the space within which a set of concepts can be carved out, and 'logical geography' refers to a particular way of carving out that space, which may correspond to how a particular community conceives of some aspect of reality. The logical topography supports the possibility of dividing things up in different ways with different tradeoffs, as different cultures divide up articles of furniture, or animals or plants in different

ways, though they are all talking about the same underlying logical topography, whether they recognise it or not. Our logical topography is concerned with the variety of relationships between a machine and its envelope of possible behaviours, or the possible sequences of envelopes if the envelope can change over time.

Overview of meta-requirements for intelligent systems

We start with high level summaries of the meta-requirements in terms of behaviour envelopes.

- **Robustness** (opposites: fragility, brittleness, unreliability)
  of a system is concerned with the ability to cope gracefully (i.e. without dramatic failure) with potential causes of hindrance, danger, difficulty, or failure such as unexpected change, signal noise, worn parts, motor or subsystem failure, poor lighting or other sensing conditions, damage to a part of the system, and a range of variability within a constrained environment of task/circumstance combinations.

  Robustness can be construed as resistance to dysfunctional perturbation of behaviour *within* an envelope of possible states and processes in which the system is behaving. Exactly what that means in each case will depend on what functional behaviour is, what the environment is, what sorts of causes of perturbation or disruption can occur, etc. Only when those have been specified can we derive the specific requirements for a system to be robust.

  Thus the requirements for a robust robot vacuum cleaner could be concerned with its ability to cope with difficult configurations of furniture as well as a range of possible mechanical deficiencies, whereas a robust robot guide for a blind person would have far more complex requirements concerned with obstacles on pavements (side-walks to Americans), traffic conditions, kerb heights, other pedestrians, and the ability to detect the need for re-charging batteries before moving too far from the nearest charging point.

  It seems that all these cases of robustness are concerned with the number of 'defensive' transitions between behaviours available for the system within its envelope of possible behaviours, and the consequences of those transitions in overcoming or preventing problems.

- **Efficiency** (opposites: inefficiency, wastefulness, resource-hungry)
  As explained above, this assumes that an aspect of the behaviour of the system is consumption of resources. Something is efficient if for most or all uses its consumption of relevant resources is low. How low it has to be can depend on whether the system is merely being compared with some sort of average, or whether there is a threshold of efficiency that has to be exceeded (efficient enough, not too inefficient, for the purpose).

- **Flexibility** (opposites: rigidity, lack of extendability)
  is not a precisely defined notion, and is sometimes used to include what we have called robustness above. But it is also sometimes used to refer to the ability to *continually extend the envelope of competence* relatively easily, without rebuilding the system or redoing low level programming.

How this extension is produced can vary. The triggers and mechanisms for extending competence beyond the old envelope will arise from new contexts of different sorts. Contexts that trigger extension can include explicit training by a teacher, instruction in some high level language (e.g. something like a human language, use of maps or diagrams or possibly some more formal and restricted notation), training by example may use the system's ability to imitate observed behaviours (a much more subtle and complex capability than many who discuss imitation, mirror neurons etc. appreciate), or the system itself doing retrospective analysis of failures or poor or failed performances (self-debugging).

The changes in competence may involve either *persistence* or *expansion* of function. In the former case no new tasks or goals can be achieved, but they can be achieved in a wider range of contexts, or in the face of a wider range of obstacles or difficulties. In the latter case (expanded function) the system acquires the ability to perform new kinds of tasks or achieve new kinds of goals.

As indicated above the process that produces the change may or may not involve explicit intervention by a user: that is part of what determines whether the system is autonomous or not, a notion discussed below.

These competence-extending developments can sometimes include combining old competences in new ways, e.g. noticing that the functions of a missing object with a particular function (e.g. screwdriver) can be provided, albeit imperfectly, by something else (e.g. a knife, or a coin), or noticing that a new type of task can be achieved by giving an old object a new function, e.g. using a big book as a step to reach a high shelf easily. (Note: some people would refer to these cases as 'creative'.)

In some cases the change make use of the pre-existing ontology used by the robot for categorising goals, objects, situations, and processes, but merely depends on it learning a new correlation, e.g. doing action A in situation S can achieve a goal of type G. Then action A will be selected in situations where it would previously not have been selected.

In other cases the changes require kinds of learning that extend the system's ontology by adding new sub-categories as a result of finding empirically, or being told, that it is useful to distinguish those sub-categories. For instance, a robot for a blind person may notice that it can define sub-categories within its action repertoire that are correlated with indications of discomfort or difficulty from the person and other sub-categories that produce expressions of pleasure or gratitude. On that basis the robot may attempt to avoid the former types of actions when there is an alternative and to select the latter types when they are available.

Or it may learn to distinguish states of the person that correlate with different preferences, e.g. wishing to move more slowly when tired or when in unrecognised surroundings. In some cases the learning involves noticing the possibility of grouping things into a new super-category. E.g. bricks, wooden blocks, large books, and small step-ladders may be grouped into a category of "things useful for getting at high shelves by stepping on them". (AI work on learning sub-categories and super-categories in various kinds of training goes back at least 30 years. The same mechanisms could work without explicit training.)

More generally, the ability to create new categories can be used in detecting patterns in sequences of things that occur, patterns expressible as *generalisations* that can be used in subsequent predictions and decisions.

The discovery of useful new sub-categories and super-categories presupposes that the existing ontology of the robot and the formalisms it uses already include means of defining the new categories. So the learning involves realisation of potential that was already implicit in the system's

knowledge, even though specific types of events were required to trigger the realisation. Insofar as the sub-categories of objects, actions, situations, etc. are related to things the robot can do and ways in which its goals can be achieved or its functions fulfilled, this kind of learning includes learning of new affordances.

Some of the kinds of flexibility described here are discussed in the forthcoming PhD thesis of Manuela Viezzer on *Autonomous Concept Formation: an Architecture-based Analysis*. This is concerned mainly with the learning of affordance concepts.

A further kind of flexibility, which requires more detailed discussion would involve modification of the set of goals or functions of the machine, either under the control of users or as a result of the system having motive generators that enable it to acquire new kinds of motives. (As discussed in Chapter 10 of *The Computer Revolution in Philosophy* (1978).

NOTE (a):
It seems clear that insofar as the notion of flexibility discussed here has several different dimensions it is probably worth subdividing this meta-criterion into several distinct meta-criteria with different labels, e.g. flexibility in functions or goals, vs flexibility of means, etc. There are many more sub-cases to be distinguished than illustrated here.

NOTE (b):
The specification of 'Target outcome (a)' for IPs and STRePS on page 3 of Colette Maloney's slides, includes two criteria of which the first, namely artificial systems that:

- "can achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; adapt, within reasonable constraints, to changing service and performance requirements, without the need for external re-programming, re-configuring, or re-adjusting."

covers both *robustness* and *flexibility* as defined here.

The second criterion might be given a label something like 'usability', 'naturalness' or even 'sociability', namely systems that:

- "communicate and co-operate with people or each other, based on a well-grounded understanding of the objects, events and processes in their environment, and their own situation, competences"

This remains a meta-criterion that only specifies actual criteria when information about the *function* or *purpose* of the system has been provided.

- **Creativity** (Opposites: Uncreative, unoriginal, dull, stupid, uninventive, unimaginative, rigid, repetitive, narrow-minded, timid(?)...)
  is not sharply distinguishable from flexibility, as those notions are normally used, but we are more likely to call a system "creative" if it has the ability to introduce greater, less gradual, extensions to the envelope of competence, and if it can do so in ways that are not triggered by external circumstances, such as problems encountered, training by a teacher, or learning from observed correlations.

  An example of competence extension could be the machine noticing the possibility of combining several objects or actions in a new way to achieve some goal or provide some new function. This typically requires the ability to construct, manipulate and compare representations of alternative combinations of objects or actions without actually combining the objects or performing the actions.

  The architectural and representational requirements for this sort of creativity are discussed in a spectrum of cases of varying complexity in this discussion paper:
  Requirements for a Fully Deliberative Architecture

  In some cases creativity may involve extending the ontology by introducing new concepts that are not definable in terms of the pre-existing ontology and formalism. This is deemed impossible by the explicit or implicit theories of many philosophers and AI researchers, but has clearly happened many times in human history and in individual humn development. The claim in Fodor's book **The language of thought**, that all such concepts are explicitly definable in terms of some innate language that is already present in a newborn

infant is entirely without foundation.

A type of robot that is to be used in a wide variety of cultures, in many different households, over many years, will require the ability to extend its ontology from time to time, not simply by adding new explicitly definable sub-categories, but by introducing a new explanatory theory in which there are new theoretical terms that are not definable using pre-existing categories, as has happened many times in the history of science and mathematics. For a robot helper this could include coming up with new theories about mental states and processes in humans which explain otherwise unpredictable changes in their behaviour, or new theories about unobservable properties of different kinds of materials or kinds of foods, which explain some of their effects, or new concepts and theories related to new developments in household goods, games, medicines, diseases, legal requirements, etc.

The ability to extend an ontology in ways that go beyond formulating definitions expressible using an old ontology has been demonstrated many times in the history of human science and culture.

Some of the requirements for this kind of ontology extension are discussed in this presentation
Getting meaning off the ground: symbol grounding vs symbol attachment/tethering

A closely related requirement is the ability to combine old competences, acquired separately, in novel ways in order to solve new kinds problems. Some examples are discussed in

Orthogonal Recombinable Competences Acquired by Altricial Species (Blankets, string, and plywood)

- **Agility** (Opposites: slowness, viscosity, inflexibility, ...)
  For more on 'agility' see the SOA document by Jason Bloomberg quoted below in the NOTES section.

  The changes of behaviour or of the envelope of behaviours mentioned in connection with robustness, flexibility, and creativity may be fast or slow. It seems that 'agility' is often used to refer to the speed with which such accommodations can occur: the faster they occur the higher the agility. Note that this is orthogonal to the quality and significance of the changes. For example, we would not wish a flexible robot vacuum cleaner to require thousands of training examples to discover the need to avoid a type of situation that causes it problems.

  A related notion (or dimension) of 'agility' would refer to the ease with which users can bring about the changes required for flexibility and creativity.

  Agility, and more generally speed, may or may not come with the cost of reduced efficiency (e.g. faster consumption of resources, increased wear and tear).

- **Autonomy**
  Autonomy comes in many shapes and forms and what is meant by 'autonomy' will typically depend on specific requirements. The words 'autonomy' and 'autonomous' are often used pretty vacuously, e.g. in connection with so-called 'autonomous agents', intended to refer to some special new kind of computing system. This ignores the fact that nowadays every operating system runs autonomously insofar as it takes many decisions without anyone explicitly telling it to. Moreover, how it takes them could be a result of an automatic learning process which adjusts scheduling or other parameters in order to optimise something. It need not be the case that the designers of the operating system specified those parameters. Moreover, users can specify all sorts of requirements, priorities, protections, privileges for other users, etc. So the users can add to the criteria used by the operating system even if they don't know exactly how those criteria will be used nor exactly what their implications will be, which could depend on what else is happening on the machine, which might be determined by other users, or by an email utility receiving messages, or a virus checker manipulating files, etc. In situations like that, neither the original designer nor the individual users will know exactly what decisions the system is taking or why it is taking them, and that is how most computers work nowadays.

  So one sort of autonomy, which we could call 'minimal autonomy', is concerned with the ability of a system to be left to get on with its task without anyone constantly overseeing and taking decisions for it. A typical automobile does not have that sort of autonomy though some tram and rail services do. Airliners may have it for parts of their flights, but not necessarily through all stages.

More significant kinds of autonomy include the system having the ability to change long term goals, short term goals, preference criteria, ways of resolving conflicts, ways of deriving means from ends, and all the other things listed in connection with robustness, flexibility and creativity. From this viewpoint each of robustness, flexibility and creativity, enhances autonomy, in different ways and to different extents.

Asimov's laws of robotics were proposed as principles for limiting the autonomy of machines, including robots, but since they were first proposed, many difficulties and objections have been pointed out.

A brief discussion can be found here Why Asimov's three laws of robotics are unethical

Note that in some cases 'autonomy' does not refer to capabilities of the system to permissions that have been granted to it. So something that is perfectly capable of taking certain decisions may be forbidden from doing so without consulting a human. That would be an extrinsic (organisational) reduction of autonomy.

to be continued/expanded.

For more on this see

- How to dispose of the freewill issue
- Four concepts of free will, two of them garbage.
- Section 10.13 of Chapter 10 of The Computer Revolution in Philosophy (1978).
  This lists some features of possible computer-based systems then states:

  > If this is not having freedom and being responsible for one's own development and actions, then it is not at all clear what else could be desired under the name of freedom.

- Daniel Dennett *Elbow Room: The Varieties of Free Will Worth Wanting*
- Daniel Dennett *Freedom Evolves*

- **Versatility** (Opposites: narrowly focused, ...)
  refers to diversity of functions, purposes, etc. This could be achieved by having completely separate subsystems, one for each function, but usually versatility implies that there is sharing of resources between different capabilities. In the case of a robot, this could also include the sharing of sensors between multiple functions, and the sharing of effectors between multiple functions. Sometimes the sharing will be sequential (you can only look in one direction, from one viewpoint, at a time) sometimes concurrent, e.g. a singer-pianist using eyes to sightread music and words at the same time, hearing your companion speaking at the same time as hearing traffic approaching, and many more.

  For more on sharing sensors, motors and other subsystems between sub-functions in a complex architecture see The Mind as a Control System (1993)

- **Being 'sensible', having common sense**
  (Opposites: lacking common sense, being stupid, forgetful, ignorant.)
  Added: 23 Jan 2007
  As John McCarthy's 80th Birthday is this year, to be celebrated at The Commonsense'07 symposium, 'Having common sense' is not a widely used system design requirement among software engineers or roboticists, though it is implicitly a requirement for the EU Challenge that triggered the writing of this document.

  Moreover, the AI researchers who defend the requirement of common sense do not always realise that it too is a meta-criterion, because the detailed requirements will be different for a nursery-school teacher, a car mechanic, a chicken farmer, a house-builder, a chef, or a researcher in mathematics. However, insofar as there is a collection of widely relevant common sense demands that arise out of acting in a 3-D physical environment, we can see that the common-sense requirement has a complex structure.

  For some examples of common-sense challenges see The Common Sense Problem Page, for example the egg-cracking problem.

Challenges close to the common sense required in a kitchen are posed in this crockery manipulation scenario. Some examples relevant to what a young child learns about manipulating objects are in Orthogonal Recombinable Competences Acquired by Altricial Species (Blankets, string, and plywood).

There are many more in McCarthy's unpublished paper 'The well designed child', and other papers on his web site. Roboticists need to heed many of the comments in that paper, including

> "Evolution solved a different problem than that of starting a baby with no a priori assumptions."
>
> "the world is not structured in terms of human input-output relations"
>
> "Animal behavior, including human intelligence, evolved to survive and succeed in this complex, partially observable and very slightly controllable world. The main features of this world have existed for several billion years and should not have to be learned anew by each person or animal."

A little understood feature of common sense is grasping that whatever is actual is but a subset of what is possible as discussed in Actual Possibilities (in KR96). This is at the heart of the notion of 'affordance' mentioned above, and the ability to plan, explain, invent stories, and create works of art.

- **Fluency: a performance requirement**
  (Opposites: jerky, lumbering, clumsy, inelegant)
  One of the features of humans, and probably many other animals, is that whereas a task may at first be performed with great care and full attention, and therefore rather slowly, it is often possible after a time for the same task to be performed quickly, smoothly and even concurrently with other tasks. Saying that it is performed smoothly implies that beginnings and ends of sub-actions are modified so that the merge in such a way as to avoid a discontinuity such as stopping after performing one sub-action and before performing the next. This is a type of improvement that does not change *what* the system can do but does change *how* it does it. An example in typing at a keyboard is the way the configuration of the hand of an expert typist when typing a particular key depends on what was typed previously and what will be typed next, so as to make the transitions smoother and faster. This kind of adjustment is possibly only when the individual actions allow quite a lot of variation in the performance.

  Exactly how this is achieved could vary according to the nature of the task. In some cases it might involve training a neural net controller to take over from a planner and plan-execution module.

  When that happens the planning and plan-execution sub-mechanisms may be able to perform additional tasks concurrently with the task 'taken over' by the neural net. In particular, complex and difficult novel tasks might be performed slowly by one mechanism while another mechanism rapidly executes some old and familiar actions.

  Other kinds of fluency could result from caching of previous results, e.g. storing generalised and flattened parse trees for frequently encountered sentence forms, or phrase forms. Some kinds of fluency are concerned with external behaviour involving control of actions, whereas others are concerned with modifying internal information processing, such as parsing, planning, checking inferences, doing calculations.

  These can all be seen as contributions to *efficiency*, or *agility* (speed) mentioned earlier, but they can be more than that, insofar as smoothness of behaviours is also increased.

- **Intentionality**
  (Opposites: lack of understanding, mere physical mechanism)
  Intentionality has been much discussed by philosophers under various labels. Many of the previous criteria presuppose it in some form or other. The key idea is that something has intentionality if it can refer to things (real or imaginary), e.g. having percepts, desires, suppositions, beliefs, memories, imaginings, goals, intentions, plans, preferences, etc. This could be summarised as having semantic competence.

  Different sorts of semantic competence can be distinguished, e.g. having the ability only to refer to what is currently being sensed or done, being able to refer to past or future events, states, processes, being able to refer to things that cannot be sensed or acted on, being able to refer to things that themselves refer or have semantic contents (second-order intentionality, third, or higher-order intentionality), etc.

Debates on the topic include

- How to tell whether something has it.
- Whether possession of intentionality is possible without having been a product of biological evolution.
- Whether ascribing intentionality to something is merely a matter of taking up a stance towards it, which may or may not have some practical use, as opposed to making a statement that can be true or false.
- Whether computer-based systems can have it or whether brain-like mechanisms are required (which can depend on the interpretation of 'brain-like'
- Which animals have it: Insects? Plants that seek water or light, or trap food? All animals or only some?
- Whether it presupposes having consciousness (whatever that means -- perhaps that's another meta-requirement, or ility: 'sensibility'?)
- Whether having consciousness presupposes having intentionality?
- How it evolved.
- Whether the mere fact that computers can obey machine code instructions and interpret machine addresses, including detecting and dealing with addresses that refer to nothing (e.g. page faults) means they have intentionality, or some kind of intentionality. (As suggested in 'What enables a machine to understand?' (1985)

(To be continued. Explain the difference between 'intensional' and various senses of 'intentional'.)

See:

Daniel Dennett, The Intentional Stance, Bradford Books, (1987)

Wikipedia on Intentionality

- **Good and bad varieties of persistence**
  (Opposites: easily discouraged, malleability, non-pig-headedness, non-steadfastness)
  'If at first you don't succeed, try, try and try again'.
  Sometimes being like that is a good trait, sometimes a serious deficiency. A full discussion would need to identify the sorts of situations in which encountering resistance, hindrances, obstacles, failures, should and should not lead to goals, sub-goals, plans, policies, etc. being abandoned, modified or replaced. Often the ability to distinguish cases where persistence is a good idea from cases where it is counter-productive requires a deep understanding of unobvious features of the situation. Sometimes it is impossible to tell and whether persistence pays off depends more on good fortune than on judgement.

.....

Other topics to be added (perhaps).

- Predictability/unpredictability
- Stability (suggested by Bernhard Sendhoff -- Honda Research)
- Variability (suggested by Bernhard Sendhoff -- Honda Research)

The next lot require meta-semantic competences.

- Prudence/discretion
- Considerateness/Kindness/Thoughtfulness
- Forgivingness/Vengefulness
- Honesty
- Adventurousness/Risk-aversion
- Impulsive/Restrained
- Courage/Boldness vs timidity
- Playfulness
- Morality?
- .....

How should we decide which meta-requirements/dimensions are relevant to a long term human-like robotics project?
Compare: how could we decide which are relevant to various kinds of domestic animals?
Compare pets vs various kinds of working animals -- guard dogs, guide-dogs, animals for riding, animals to help with physical labour, animals that can get to places more easily and quickly, etc., animals that help with herding other animals, animals that perform and entertain....

NOTE on software 'ilities' (Added 22 Jan 2007)

David Vernon informed me that some of the topics discussed here have also been discussed in the software engineering research community, under the heading of 'ilities' and other generic labels. Examples are these web sites:

- Architecture Requirements are Ilities
  (The 'Software Architecture Notes' web site: includes a 'Starter List of "Ilities"')

  > An "ility" is a characteristic or quality of a system that applies across a set of functional or system requirements. So, performance is an "ility" because it is applied against some of the functional or system requirements. Anything that can be expressed in the form "for a set of functional or system requirements, the system must fulfil them this way (this fast, this reliable, etc.)" is an "ility."
  >
  > .....
  >
  > It is important to find as many of these and describe them as accurately and as early as possible. Since they describe ways that sets of functional requirements must be satisfied, they are effort multipliers to develop. So, for example, if a set of functions have to be secured, then the effort to secure a single function must be multiplied across each of the functions to be secured.

- Software's Best Kept Secret: 'ilities', by Luc K. Richard
  (Distinguishes external and internal ilities.)

  > Many developers make the mistake of thinking that quality attributes -- commonly referred to as nonfunctional requirements, technical requirements or ilities -- are somewhat superfluous. Convinced that these nonfunctional requirements are not as critical to the end product as functional requirements, ilities are rarely documented or even understood.

- Wikipedia's list of 'ilities'
  (Also referred to as 'non-functional' requirements)

It turns out that giving google both "software" and "ilities" produces a very large number of documents.

We have not yet found a characterisation of the ilities like the one offered for meta-requirements in this discussion paper, namely in terms of functions that transform behaviour envelopes in a manner that can be specified at a level of abstraction that is independent of the actual behaviours. However, my specification of those transformations is still very informal.

Non-functional?
The characterisation of these requirements as 'non-functional' by some theorists seems to me to be mistaken. They are highly important for functionality, but they are higher-level characterisations that determine the specific form of functionality only in combination with additional information,

> just as the map function given a list returns a new list, but only if provided with an extra argument, e.g. given a list of numbers AND a function, such as sqrt,
>
> ```
> map(list, sqrt)
> ```
>
> returns a list of the square roots of the original numbers. The fact that 'map' requires another function as argument does not stop it being a function itself. It is merely a second order function.

The quotation from the 'Software Architecture Notes' web site above, seems to be making a similar point, namely that the ilities (or what we have called 'meta-requirements') add further specification to a set of functional requirements (e.g. specifying the 'way' the requirements should be met).

A difference between the goals of this document and the software engineering discussions is that we have tried to discuss meta-criteria (ilities) that could be equally relevant to biological organisms and engineering artifacts, since the EU FP7 Challenge 2 is in part about biologically inspired systems, but not inspired at the level of *mechanisms*, as the phrase 'biologically inspired' often indicates.

NOTES

1. For further discussion of meta-concepts, or higher-order concepts, see this draft discussion paper Spatial prepositions as higher order functions: And implications of Grice's theory for evolution of language.

---

2. **Note added 20 Jan 2007**
   Since deciding to use the label 'meta-requirement' we have discovered that others already use it, sometimes in a different way from the above (e.g. to specify requirements for requirements) and sometimes in a similar way (i.e. to specify schematic requirements that only determine specific requirements in relation to *other* behavioural requirements).
   An example is in this paper on Service-Oriented Architecture (SOA):
   OA Quality and Governance: Satisfying the Metarequirement of Agility
   By Jason Bloomberg
   Posted: Aug. 24, 2006

   "**Change Time Quality**
   Traditional software quality management essentially consists of design time and deployment time activities. Basically, given the requirements, make sure that the software is as defect-free as possible given budget and schedule constraints, and then continually monitor the working software to make sure that it meets the requirements set out for it when you deploy it. That basic approach to quality is fine for organizations that know in advance what their requirements are, when those requirements are stable, and when the goal is simply to build software that meets those requirements.

   Such assumptions, however, are frequently false -- in many cases, requirements aren't fully developed and they change over time. Typically, one true goal of software is to respond to changes in requirements without extensive additional rework. SOA is a particularly effective approach in such situations, and the broad recognition that the build-to-today's-requirements approach to software is no longer effective is one of the primary motivations for SOA."

REFERENCES

http://www.scielo.cl/pdf/rfacing/v13n1/art08.pdf

Hernan Astudillo, Five Ontological Levels to Describe and Evaluate Software Architectures, in *Rev. Fac. Ing. Univ. Tarapaca, vol.* 13 no. 1, 2005, pp. 69-76

(Partly overlaps with this paper - not yet studied closely.)

Paper by Joel Moses (added 16 Nov 2008)

At The Workshop on Philosophy and Engineering (WPE'2008) in London, November 10-12 2008, Joel Moses presented a paper 'Toward an Ontology for Systems-related Terms in Engineering and ComputerScience', closely related to this one. His two page extended abstract is included in the workshop proceedings (Unfortunately a MSWord 'doc' file. However, the freely available OpenOffice package converts that to PDF.)

See additional references in the main text.

More to be added

Maintained by Aaron Sloman
School of Computer Science
The University of Birmingham