RockEU2

Robotics Coordination Action for Europe Two

Grant Agreement Number: 688441

01.02.2016 – 31.01.2018

Instrument: Coordination and Support Action

Cognition-Autonomy Framework

David Vernon, University of Skövde, Sweden

Markus Vincze, Technische Universität Wien, Austria

Deliverable D3.4

Lead contractor for this deliverable:	University of Skövde
Due date of deliverable:	July 31, 2017
Actual submission date:	August 21, 2017
Dissemination level:	Public
Revision:	1.0

History of Changes

Name	Status	Version	Date	Summary of actions made

Executive summary

The goal of Task 3.4 was to investigate the possibility that the terminology used when discussing autonomous systems may be a more natural way for users and robot developers to express their needs, compared with the terminology of cognitive systems. This deliverable presents the results of this investigation. It first considers what is meant by autonomy, highlighting the interpretations that are most relevant to cognitive robotics. It presents a taxonomy of the characteristics of autonomous systems and, in so doing, sets out the terminology that can be used to describe various aspects of autonomy. It concludes that there is significant merit in adopting the language of autonomous systems when specifying requirements for cognitive robots. This conclusion is based on a validation exercise to use the terminology in the specification of a meta use-case that has been derived from Deliverable D3.1 Industrial Priorities for Cognitive Robotics. This validation is part of a larger exercise to link together the results from Work Package 3; for a summary of this exercise, please refer to Appendix I.

Content

1. Introduction

In specifying the requirements for cognitive robots, we need an appropriate terminology, one that is intuitive to use, that resonates with the language used by end-users and system developers, and that can be used effectively to drive the subsequent processes of system analysis, specification, and design. The goal of Task 3.4 was to investigate the possibility that the terminology used when discussing autonomous systems may be a more natural way for users and robot developers to express their needs, compared with the terminology of cognitive systems. The language we use when discussing autonomy tends to focus on *what* the system does and *why*, whereas discussions of cognitive systems often tends to revolve around *how* they do these things. This may not be helpful to roboticists who are concerned with using cognition to achieve a desired behaviour or action and are less concerned with the mechanisms by which this is achieved. The goal then was to explore the utility of casting requirements in terms of autonomy and then mapping these to requirements in terms of cognitive systems.

This deliverable first considers what is meant by autonomy, highlighting the interpretations that are most relevant to cognitive robotics. It presents a taxonomy of the characteristics of autonomous systems and, in so doing, sets out the terminology that can be used to describe various aspects of autonomy. *It concludes that there is significant merit in adopting the language of autonomous systems when specifying requirements for cognitive robots: the terminology used when discussing autonomous systems is indeed a more natural way for users and robot developers to express their needs, compared with the terminology of cognitive systems.* This conclusion is based on a validation exercise to use the terminology in the specification of a meta[1] use-case that was derived from Deliverable D3.1 Industrial Priorities for Cognitive Robotics. This validation is part of a larger exercise to link together the results of Work Package 3; details of which can be found in Appendix I.

These results provide a new way of expressing the requirements of industrial robotics for cognitive capabilities, one that is more natural for user and developer but which can still be mapped to the underlying computational model required to deliver them. The impact may be significant in that it makes it easier for industrial roboticists to identify the needs of their systems and make technology transfer more efficient, effective, and less prone to mismatches in expectation and reality.

2. The many views on autonomy

Autonomy is a difficult concept to tie down [1] and there are several perspectives on what it means [2]. Nonetheless, most people agree that autonomy reflects the degree of self-determination of a system, i.e. the degree to which a system's behaviour is not determined by the environment and, thus, the degree to which a system determines its own goals [3, 4, 5, 6].

More than twenty types of autonomy can be distinguished [7]. For example, you will see references to the following, among others.

> Adaptive autonomy, adjustable autonomy, agent autonomy, basic autonomy, behavioural autonomy, belief autonomy, biological autonomy, causal autonomy, constitutive autonomy, energy autonomy, mental autonomy, motivational autonomy, norm autonomy, robotic autonomy, shared autonomy, sliding autonomy, social autonomy, subservient autonomy, user autonomy.

The different types of autonomy can be categorized in several ways, e.g. under the headings of robotic autonomy and biological autonomy [8]. In the current context, robotic autonomy is more relevant and we will focus our discussion on this category. However, we will refer later to an aspect of biological autonomy that has a particular bearing on cognitive robotics.

[1] It is a meta use-case because it is more generic in its description than would be a use-case for a real target application.

2.1. Strength and degree of autonomy

In robotics, it can be useful to categorize the capabilities of a robot on the basis of (a) its ability to deal with uncertainty in its environment and (b) on the extent to which a human operator assists the robot in pursuing a task and achieving some goal (see Figure 1).

The ability to deal with uncertainty in carrying out a task is sometimes referred to as *task entropy* [9]. At one end of the task entropy spectrum there are tasks that are completely pre- specified. There is no uncertainty at all about the task, the objects, and how to go about achieving the goal. Everything is fully known. This is a low-entropy task. At the other end of the spectrum there is significant uncertainty about the task and there is a lot of unpredictability about what objects are present, where they are, what they look like, and what is the best way of achieving the goal of the task. This is a high-entropy task. We use the term *strength of autonomy* to denote the extent to which an autonomous system can deal with this unpredictability: strong autonomy indicates that the system can deal with considerable uncertainty in the task whereas weak autonomy indicates that it cannot.

On the other hand, the extent to which a human assists the robot reflects the degree of automation realized by the robot. The term *degree of autonomy* is often used to indicate the relative balance of automatic and human-assisted operation. At one end of the scale, we have entirely manual operation. This corresponds to a tele-operated robot, i.e. a robot that is controlled completely by a human operator, possibly mediated through a computer system, and typically from some distance away. At the other end of the scale, we have completely automatic operation, i.e. the robot operates entirely on its own, with no assistance or intervention by a human operator. The relative balance between the two ends of the spectrum is captured by the related terms subservient autonomy, adjustable autonomy, shared autonomy, and sliding autonomy [10].

The strength of autonomy is sometimes referred to as self-sufficiency: the capability of a system to take care of itself. The degree of autonomy is also referred to as self-directedness: freedom from outside control [11].

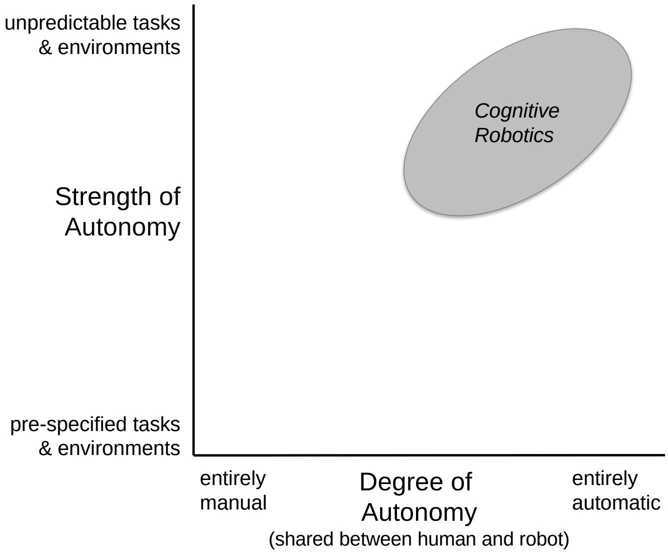
unpredictable tasks & environments
Cognitive Robotics
Strength of Autonomy
pre-specified tasks & environments
entirely manual
Degree of Autonomy
(shared between human and robot)
entirely automatic

Figure 1: An autonomous agent — a person or a robot — can be situated in a two-dimensional space spanned in one dimension by the amount of unpredictability in the task and the working environments and in the other dimension by the degree of human assistance that is required. These two dimensions are the strength of autonomy and the degree of autonomy. This figure was adapted from one that appears in [9].

As the scale of the scientific and technological challenge of creating fully autonomous robot solutions becomes more evident, the virtue and necessity of shared autonomy becomes apparent and topical. For example, at the time of writing it is a current Research Topic in the open source journal Frontiers [12] where it is noted that “fully automated behaviour is still a long way ahead and we therefore rely on some level of human supervision creating shared autonomy systems which leverage the strengths of both human and robots. In Shared Autonomy we either interact, cooperate, communicate or at least have to find actions safely not interfering with the other. Shared Autonomy as a concept describes how all the agents can remain autonomous, following overall their own intentions and goals, but at the same time deal with coordination of activities and resolution of possible conflicts”.

Shared autonomy not only involves a human assisting a robot to some extent — small or large — in situations where help is needed but it also applies in the case where a robot assists a human to some degree, ranging from passive cooperation (or facilitation), to active helping, to full-blown collaboration where the intentions and goals are shared between both human and robot. This aspect of shared autonomy differentiates it from adjustable autonomy or sliding autonomy and takes it into a realm of sophisticated behaviour and cognitive capability that is presently far beyond our scientific and technological capabilities, even if it provides a convenient encapsulation of the long-term goal of robot autonomy.

To avoid confusion, we will use the term adjustable autonomy when we mean the weaker degree of autonomy (in which the task is shared by the robot and human, with the human providing assistance when necessary) and we will use the term shared autonomy when we mean the stronger degree of autonomy (where the robot cooperates, helps, and collaborates with the human).

2.2. Behavioural autonomy and constitutive autonomy

The different types of autonomy can be categorized in other ways, e.g. behavioural autonomy and constitutive autonomy [2, 13]. This categorization is useful because, as we will see, the associated terminology mirrors the terminology that is used when discussing robust large-scale software systems.

Behavioural autonomy is concerned with the external behaviour of the system: the extent to which the agent sets its own goals and its robustness and flexibility in achieving them as it interacts with the world around it, including other cognitive agents. For the most part, this is what was discussed in the previous section.

Constitutive autonomy is concerned with the internal organization and the organizational processes that keep the system viable, maintaining itself as an identifiable autonomous entity. Maturana and Varela, whose work provided the inspiration for the enactive view of cognition, define autonomy as “the condition of subordinating all changes to the maintenance of the organization” [14].

For biological autonomous agents, as well as bio-inspired artificial agents, the issue of autonomy is one of survival in the face of precarious conditions, operating in an uncertain possibly- dangerous constantly-changing environment. To do this, it must keep itself intact as an autonomous system, both physically and organizationally as a dynamic self-sustaining entity. The self-maintenance of autonomy is a crucial aspect of enactive cognitive agents [15], continually re- pairing damage to themselves. Since it is better if the agent can avoid damage in the first place, cognition, as a prospective modulator of perception and action, is one of the primary mechanisms at the agent’s disposal [13], allowing it to anticipate the need for action and the outcome of that action. From this perspective, autonomy, aided by cognition, is the self-maintaining organizational characteristic of living creatures that enables them to use their own capacities to manage their interactions with the world in order to remain viable [16]. In other words, auton- omy is the process by which a system manages — self-regulates — to maintain itself as a viable entity despite the precarious conditions with which the environment continually confronts it.

So what has this to do with cognitive robotics? The answer becomes evident when we examine autonomic computer systems, the topic to which we now turn.

2.3. Autonomic systems

The ultimate goal for many people in computer technology is to produce a software-controlled system that you can simply turn on and leave to its own devices, knowing that if anything unforeseen happens, the system will sort itself out. This very desirable capability is often referred to as autonomic computing, a term introduced by IBM vice-president Paul Horn in a keynote address to the National Academy of Engineers at Harvard University in March 2001. He defined autonomic computing systems as systems that can manage themselves, given high-level objectives from administrators [17]. Thus, autonomic computing is strongly aligned with the concept of subservient autonomy which we discussed above. The term autonomic computing was inspired by the autonomic nervous system found in mammals, i.e. the part of the nervous system that operates automatically to regulate the body's physiological functions such as heart-beat, breathing, and digestion. Thus, autonomic computing is also strongly aligned with self-regulatory biological autonomy, in general, and homeostasis and allostasis, in particular. Autonomic computing systems aim to exhibit several operational characteristics, including the ability to be self-configuring, self-healing, self-optimizing, and self-protecting [18, 19].

3. The terminology of autonomous systems

With that very brief summary of autonomy in place, we move on now to the main goal of this document: to classify the functional characteristics of autonomous systems, to identify the terms that are used to refer to these characteristics, and to then show that this terminology can be deployed in a straightforward and expressive manner to describe the functional requirements of cognitive robots without reverting to the specialized language of cognitive science. This provides users and industrial developers with an alternative lexicon which may be more natural to use when focussing on the applications of cognitive robots and, thereby, aiding the adoption of cognitive robotics in industry. It then remains to then map these requirements, rendered in the terminology of autonomy, to the cognitive mechanisms by which they can be satisfied.

3.1. Characteristics of autonomous systems

A quotation from Thórisson and Helgasson will allow us to begin our characterization of autonomous systems (AS).

> "Autonomous systems automatically perform tasks in some environment, with unforeseen variations occurring in both, through some type of automatic learning and adaptation that improves the system's performance with respect to its high-level goals" [20].

Thus, an AS performs tasks, the tasks have unforeseen variations, and the environment in which they perform them have unforeseen variations. An AS learns how to adapt to these variations and improves its performance. Thórisson and Helgasson continue:

> "Learning and adaptation in this context refer to the ability of a system, when facing situations with some degree of similarity to those already experienced, to consistently alter its responses based on what worked in the past, and improve the system's responses such that over time they become incrementally better in respect to the active goals or utility function(s)" [20].

Thus, an AS can assess a situation, recognize that it is different from what it has experienced previously, and formulate an strategy (or action policy) to deal with it. That strategy may just be an incremental variation on its previous action policy or it might be a completely new strategy based on some observed pattern in the current repertoire of action policies.

An AS can request help if it does not know how to complete a task [10]. This help can come in the form of information — knowledge or know-how — from any available resource. It might also come in the form of physical assistance from a human or another robot.

An AS, especially one that has a high degree of autonomy, has situation awareness and a "forward thinking capability to comprehend the repercussions of the environmental cues ... and to anticipate and proactively prepare (through planning or adaptive behaviour) for foreseeable situations, in addition to executing sequences of actions that are hardwired during design time" [21]. Thus, an AS is capable of anticipating the need for actions and the outcome of its actions.

An AS can discriminate between what is important and what is not important when carrying out a task. In other words, it is able to pay selective attention and, furthermore, it has some expectations with regard to its upcoming tasks to focus that attention and steer its actions [20].

Anderson et al. [22] state that an AS, like a human, should notice when some anomaly occurs, assess it, and guide a response. They refer to to this as a NAG — Notice, Assess, Guide — loop. It is a basic form of metacognition, or the metacognitive loop (MCL), that is typical of autonomous systems. MCL is an important way for an AS to deal with surprises and learn from them. Equivalently, it reflects an ability to generate expectations, monitor these expectations, note failed expectations, assess the causes, choose appropriate responses, and learn from these situations by modifying its expectations. It should be able to explain what went wrong, i.e. why an expectation was not fulfilled, and why. An AS, then, knows what it is supposed to be doing, can deal with mistakes, knows when it is not being successful, and can either ask for help or figure out an alternative approach on its own.

So far, all of the characteristics that we've discussed so far are focussed mainly on the manner in which the AS interacts with the environment, including other agents. In a sense, they reflect the issues of behavioural autonomy. The complementary aspects of constitutive autonomy are also relevant and yield a related set of characteristics, many of which reflect the attributes of autonomic systems. They include the ability to self-monitor, self-regulate, self-repair, self- describe [19]. Crowley et al. explain that the prefix *self-* is different from *auto-* in that it entails that the autonomic ability is provided using explicit declarative or symbolic descriptions of the system [19]. Self-monitoring involves the observation of internal state, including quality-of-service metrics, such as reliability, precision, rapidity, and throughput. It maintains a model of its own behaviour to estimate confidence for its outputs. Self-regulation involves the adjustment of internal parameters to ensure these quality-of-service commitments. Self-repair concerns the reconfiguration of the system in response to changes in external requirements or operating conditions. It involves three phases of error detection, error diagnosis, and error correction. Self-description provides a description of the system's internal state.

Kephart and Chess, recounting the introduction of the term *autonomic computing* by IBM, use slightly different complementary terms under the general umbrella of self-management, the identify self-configuration, self-optimization, self-healing, and self-protection as the essence of autonomic computing systems [17]. Self-configuration is the ability of an AS to configure itself automatically in accordance with high-level policies representing, e.g. user objectives. Self- optimization is the ability to continually seek ways to improve operational performance. Self- healing refers to the ability to detect, diagnose, and repair localized problems. Self-protection has two aspects: (a) the ability to defend the system as a whole to malicious attacks or cascading failures, and (b) the ability to anticipate problems based on early reports from sensors to prevent system-wide failures (remember: Kephart and Chess are discussing autonomic computing and so their focus is on software systems).

The complementary terminology of autonomic computing reflects the source of the AS goals. Autonomic processes focus on system operation but humans provide the policies, i.e. the goals and constraints that govern their actions. Kephart and Chess note two concerns about these goals: that it is important to ensure that goals are specified correctly in the first place, and that the system behaves responsibly when they are not. They suggest that autonomic systems need to protect themselves from input goals that are inconsistent, implausible, dangerous, or unrealizable with the resources at hand. Even so, the dependence on human-specified policies is crucial. As Bradshaw et al. remark:

> "Policies are declarative constraints on system behaviour that provide a power means for dynamically regulating the behaviour of components without changing code nor [sic] requiring the cooperation of the components being govered" [11].

Kephart and Chess note that while the term autonomic computing was chosen deliberately to have biological connotations with an organisms autonomic system, it also recognizes the "enormous range in scale" of systems that exhibit the same characteristics of self-governance (i.e. autonomy), from the molecular level to the economic markets, societies, and global socio- economic systems [17].

3.2. Terminology

From the foregoing section, we can extract the key terms that characterize the behaviour and operation autonomous systems. These are listed in Table 1. Our contention is that these terms for an appropriate lexicon with which to specify the requirements of a cognitive robot, an exercise which we undertake in the next section.

Behaviour	**Operation**
Perform tasks Adapt to unforeseen variations in tasks Adapt to unforeseen variations in the environment Improve performance with respect to active goals Accept external policies Form new action policies Ask for and accept help Seek additional information Have situation awareness Anticipate the need for actions Anticipate the outcome of actions Pay attention Form expectations Detect anomalies Know what to do and what is being done Learn from the unexpected	Self-monitor Self-regulate Self-repair Self-describe Self-configure Self-optimize Self-protect

Table 1: Terminology of Autonomous Systems

4. A validation exercise

4.1. Industrial priorities for cognitive robotics

Based on the results of a survey of industrial developers to determine what they and their customers require from a cognitive robot, RockEU2 Deliverable 3.1 Industrial Priorities for Cognitive Robotics presented a series of eleven functional abilities needed for cognitive robots. For convenience, we summarize these here.

1. **Safe, reliable, transparent operation**. Cognitive robots will be able to operate reliably and safely around humans and they will be able to explain the decisions they make, the actions they have taken, and the actions they are about to take.
2. **High-level instruction and context-aware task execution**. Cognitive robots will be given tasks using high-level instructions and they will factor in contextual constraints that are specific to the application scenario when carrying out these tasks, determining for themselves the priority of possible actions in case of competing or conflicting requirements.
3. **Knowledge acquisition and generalization**. Cognitive robots will continuously acquire new knowledge and generalize that knowledge so that they can undertake new tasks by generating novel action policies based on their history of decisions. This will allow the rigor and level of detail with which a human expresses the task specification to be relaxed on future occasions.
4. **Adaptive planning**. Cognitive robots will be able to anticipate events and prepare for them in advance. They will be able to cope with unforeseen situations, recognizing and handling errors, gracefully and effectively. This will also allow them to handle flexible objects or living creatures.
5. **Personalized interaction**. Cognitive robots will personalize their interactions with humans, adapting their behaviour and interaction policy to the user's preferences, needs, and emotional or psychological state. This personalization will include an understanding of the person's preferences for the degree of force used when interacting with the robot.

6. **Self-assessment**. Cognitive robots will be able to reason about their own capabilities, being able to determine whether they can accomplish a given task. If they detect something is not working, they will be able to ask for help. They will be able to assess the quality of their decisions.
7. **Learning from demonstration**. Cognitive robots will be able to learn new actions from demonstration by humans and they will be able to link this learned knowledge to previously acquired knowledge of related tasks and entities.
8. **Evaluating the safety of actions**. When they learn a new action, cognitive robots will take steps to verify the safety of carrying out this action.
9. **Development and self-optimization**. Cognitive robots will develop and self-optimize, learn- ing in an open-ended manner from their own actions and those of others (humans or other robots), continually improving their abilities.
10. **Knowledge transfer**. Cognitive robots will be able to transfer knowledge to other robots, even those having a different physical, kinematic, and dynamic configurations and they will be able to operate seamlessly in an environment that is configured as an internet of things (IoT).
11. **Communicating intentions and collaborative action**. Cognitive robots will be able to communicate their intentions to people around them and, vice versa, they will be able to infer the intention of others, i.e. understanding what someone is doing and anticipating what they are about to do. Ultimately, cognitive robots will be able to collaborate with people on some joint task with a minimal amount of instruction.

For further details, please refer to the deliverable [23, 24].

Although these functional abilities help greatly in defining the nature of cognitive robots, it is not easy to map them directly to a cognitive architecture that can be used to drive the development of a cognitive robot and, in particular, to identify the computational mechanisms required by that cognitive architecture. To do that, we require a more detailed set of requirements. To provide those, we now proceed to derive from this list a meta use-case using the terminology of autonomous systems. This meta use-case determines the required cognitive functional requirements of the robot which, in turn, determine the AI tools and techniques needed to design an appropriate cognitive architecture. It is a meta use-cases because it is more generic in its description than would be a use-case tied to a real target application.

A separate deliverable, Deliverable D3.2, presents a catalogue of implemented and accessible cognitive systems capabilities by identifying the AI tools and techniques that can be deployed today to satisfy the cognitive functional requirements. The encapsulation of these tools and techniques in a cognitive architecture is taken up in Deliverable 3.3 and the software engineering implications of realizing such an architecture are addressed in Deliverable D3.5. Refer again to Appendix I to see how these several strands — specifically the five tasks and deliverables of Work Package 3 — are woven together.

4.2. Meta use-case: a personal robot shopping assistant

In the following use-case, the terms in Table 1, or variants of them, are set in italics. We do not claim that this meta use-case is complete: a full application specification would surely require more detail. The main point to be made in this section is that it can be expressed very naturally in the terminology of autonomy, particularly the terminology associated with behaviour. Even though the meta use-case is not complete, it does reflect all eleven industrial priorities, each of which is flagged in the following as a superscript number.

> The personal robot shopping assistant will to be used by shops to provide elderly customers with support and help with the shopping by guiding them around the shop, avoiding crowded aisles, carrying their goods, taking goods from shelves, informing the customer about goods that the customer wants to buy, suggesting goods that are usually bought with the ones already picked, providing guidance about staying within budget, and helping them unload and pack goods at the checkout. The robot shopping assistant and the customer interact with each other using verbal communication. The robot's movement can be controlled by the customer in the same way as a normal shopping cart by pushing but it provides servo assistance that is adapted to the individual customer’s needs.

The robot should *pay attention* to the customer and it should *perform tasks* requested by him or her. The customer should be able to express the requests in simple spoken terms.[2]

Most of these tasks correspond to a standard repertoire that have been pre-defined using *external policies*. However, if the robot perceives the customer to be following a new pattern of shopping, it should *form new action policies* that make these new tasks simpler, more efficient, or more enjoyable.[3,5,9]

Similarly, the robot should attempt to detect any patterns in the nature of the items on the shopping list, e.g. based on its knowledge of household chores or baking recipes*, form some expectation* of what is required, and suggest items that might have been omitted.[3]

After a few shopping trips with the same customer, the robot should *form expectations* about the customer's preferences.[5] When navigating the aisles in the shop, the robot should develop an *awareness of the situation*, avoiding aisles that are busy and perhaps uncomfortable for the customer, suggesting they go to get another item, and explaining why.[1,2,4,11]

If the required items are not in the expected location, the robot should adapt and *either ask for help* from other customers or a human shop assistant.[4,6] Alternatively, it should *seek additional information* from the customer.[3,6]

If the robot drops the item it has grasped, it should *know what it was doin*g, realized that something *unforeseen* happened, *adapt* to pick it up, placing it in the shopping trolley.[4,6]

When selecting loose fruit or vegetables, it should avoid items that are odd-looking or *anomalous*. It should only select and pick items if *anticipates the picking action* will be successful. If items fall *unexpectedly* after it removes them from the pile, it should realize that this should not have happened and create *a new policy* to improve its performance for selecting items in future.[4,6]

Similarly, when selecting similar products from a shelf, it should *know* to read the product description and price to the customer without being asked to do so.[11]

When searching for an item, it should *explain* to the customer *what it is doing*.[1,11]

All selected items should be placed in the trolley in an appropriate place: the robot should *anticipate* that placing a large bottle of water on a carton of eggs may break them and avoid doing so.[8]

Similarly, when transferring the goods from the basket to the checkout, it should *anticipate* the best way to pack the items and select the items to be transferred accordingly.[8]

If the customer elects to use a self-service checkout, the robot should be able to follow all the instructions meant for humans.[2] If it is unable to do so, it should *ask a human* shop assistant to demonstrate how it should be used.[7,9]

When the robot has finished helping a customer, it should upload everything it has learned to a central knowledge repository to which other robot shopping assistants have accessess.[10]

5. From autonomy to AI tools & techniques and cognitive architectures

Our ultimate goal in Work Package 3 is to further the deployment of cognitive robotics in industry. To attain that goal we need to identify the necessary AI tools & techniques based on the requisite cognitive functional requirements which, in turn, are derived from meta use-cases that reflect industrial priorities and that are expressed in the terminology of autonomous systems.

However, we need to be cautious: it is not necessarily a trivial task to assemble a complete cognitive system even if all the AI tools and techniques are in place. Heinz von Foerster argues that the constituents of a cognitive system cannot be separated into distinct functional components:

> "In the stream of cognitive processes one can conceptually isolate certain components, for instance (i) the faculty to perceive, (ii) the faculty to remember, and (iii) the faculty to infer. But if one wishes to isolate these faculties functionally or locally, one is doomed to fail. Consequently, if the mechanisms that are responsible for any of these faculties are to be discovered, then the totality of cognitive processes must be considered." [25], p. 105.

Thórisson and Helgasson make a similar point while offering a concrete way of making progress:

> "We see little reason to believe that the collective work of the field as a whole, offering isolated cognitive abilities targetting tasks often with greatly limited scope can somehow be fused to give rise to systems possessing general intelligence ... [but] the most promising work for creating machines with general intelligence belong to the area of cognitive architectures" [20].

Consequently, in what follows we cast the cognitive functional requirements in terms that are easily mapped to cognitive architectures, which can then draw as necessary on the available AI tools and techniques. Deliverable D3.2 expands further on the rationale to focus on a complete cognitive architecture rather than a list of AI tools and techniques.

We turn now to the task of identifying the cognitive functional requirements associated with the behaviour-oriented autonomous system terminology set out in Table 2 and used in the meta use-case above. We do so simply by listing the functional requirements alongside the corresponding AS term.

	Perception	Declarative knowledge & memory	Procedural knowledge & memory	Planner & plan executive	Reasoning & inference mechanisms	Meta-cognition	Environment model	Internal simulator	Goal representations	Declarative learning	Procedural learning	Attention mechanism	Realtime action controller
Perform tasks	x	x	x	x	x		x		x			x	x
Adapt to unforeseen variations in tasks	x	x	x	x	x	x	x		x	x	x	x	x
Adapt to unforeseen variations in the environment	x	x	x	x	x		x	x		x	x	x	x
Improve performance with respect to active goals		x	x	x	x	x	x	x	x		x	x	x
Accept external policies		x	x						x				
Form new action policies		x	x		x	x		x	x	x	x		
Ask for and accept help		x	x			x			x	x	x	x	x
Seek additional information		x	x			x			x	x	x		
Have situation awareness	x	x	x		x		x					x	
Anticipate the need for actions		x	x		x		x	x	x				
Anticipate the outcome of actions		x	x		x		x	x	x				
Pay attention	x	x	x					x	x			x	x
Form expectations		x	x		x			x	x	x			
Detect anomalies	x	x	x		x	x	x	x				x	
Know what to do and what is being done		x	x	x					x				
Learn from the unexpected	x	x	x		x	x	x		x	x	x	x	

Table 2: Autonomous systems terminology *vs*. cognitive functional requirements.

Note that this approach to specifying the functionality of a cognitive architecture functionality by deriving it from specific application requirements has been dubbed *design by use-case* in contrast to *design by desiderata* based on the requirements of a general cognitive architecture [26]. Nevertheless, it is instructive to cross-check these functional requirements with those identified by Thórisson and Helgasson in their own exercise to determine the required components of a cognitive architecture based on their characterization of autonomy:

1. An attentional mechanism to select which sensory data to process and how deeply.
2. An internal simulation mechanism to facilitate prediction.
3. A reasoning and inference mechanism.
4. A means to construct models of the environment that can be used for coupled reasoning and prediction.

5. Diverse mechanisms for memory.
6. Perception.
7. A planning mechanism.

Thórisson and Helgasson highlight in particular four themes: realtime operation, resource management, learning, and meta-learning. Realtime operation requires a general time management mechanism and an ability to act in synchronization with the system's surroundings. Resource management is a mechanism to prioritize processing and applies to all data in the system, including tasks and goals. Learning — the general capability of a system to improve its performance over time — comes in many forms, e.g. supervised, reinforcement, and unsupervised learning, all of which may be necessary. Learning and knowledge go hand in hand so the architecture should also have the appropriate forms of knowledge representation. Meta-learning means the ability to make changes to itself, both in respect of how it learns and how it controls its own inner workings, resulting in an improvement in its own performance. Meta-learning is sometimes referred to as meta-cognition which, typically, also includes mechanisms for resource management. Consequently, to the list above, we can add:

8. A time management mechanism to facilitate realtime interaction with the environment.
9. A mechanism for managing system resources.
10. Diverse forms of learning.
11. Diverse forms of knowledge representation.
12. A mechanism for meta-cognition: the ability to monitor and adaptively improve the system's own operation.

All twelve are already included in Table 2 as a consequence of the meta use-case analysis. Of course, the literature is not short of articles identifying the desirable features of cognitive architectures, e.g. [27] and [28]), but we restrict ourselves here to those suggested by Thórisson and Helgasson because they derive from a specific exercise to link the characteristics of autonomy to the functional requirements of cognition.

6. Summary and Conclusions

Two factors are necessary to specify the autonomy of a cognitive robot: the degree of autonomy (including the level of self-determination) and the strength of autonomy. Cognitive robots do not have to have a very high of autonomy to be useful and, indeed, this may not desirable: the ability to take instruction from humans and operate safely when interacting with them is more important. This flexibility is related to the concepts of adjustable, sliding, and, in one sense at least, shared autonomy.

A robot with some degree of autonomy is capable of several things. It is capable of independently achieving certain tasks. It knows when it needs help. It recognizes when help is being offered. It is able to seek help at the appropriate time, taking into consideration the fact that the user or operator may take time to assess the situation and respond. It is able to accept help and use that help. It is able to learn while help is being provided. A robot that exhibits strong autonomy has the ability to operate (reason and make decisions) in the presence of uncertainty and in the absence of information, both in terms of its tasks and its operating environment. It learns as it does so, adapting its action policies so that it performs better in the future and better equipped to deal with unforeseen events. A robot that is simple to instruct has some mechanism for the user or operator to specify the goals of a task or the task (or both) to be performed at some level of abstraction. Robots with a high degree of autonomy can accommodate roughly-stated possibly vague instructions, resulting in less effort on the part of the user or operator.

In this document, we have set out the terminology that is used when describing the behaviour of autonomous systems. We don't claim that the terminology is complete but it is sufficient to demonstrate its usefulness for specifying the required functionality of a cognitive robot. We have done this with the aid of a meta use-case, using the autonomous systems terminology to specify the use case. This use-case also reflects the eleven industrial priorities for cognitive robotics set out in Deliverable D3.1. Finally, we have mapped the terminology to a set of cognitive functional requirements which will be used to cross-reference the cognitive abilities addressed in Deliverable D3.2 and the cognitive architecture requirements in Deliverable D3.3 (again, please refer to Appendix I for a summary of how all these Work Package 3 tasks and deliverable fit together).

7. References

[1] M. A. Boden. Autonomy: What is it? BioSystems, 91:305–308, 2008.

[2] T. Froese, N. Virgo, and E. Izquierdo. Autonomy: a review and a reappraisal. In F. Almeida e Costa, L. Rocha, E. Costa, I. Harvey, and A. Coutinho, editors, Proceedings of the 9th European Conference on Artificial Life: Advances in Artificial Life, volume 4648, pages 455–465, Berlin Heidelberg, 2007. Springer. doi: 10.1007/978-3-540-74913-4 46.

[3] T. Ziemke. The 'environmental puppeteer' revisited: A connectionist perspective on 'autonomy'. In Proceedings of the 6th European Workshop on Learning Robots, Brighton, UK, August 1997.

[4] T. Ziemke. Adaptive behaviour in autonomous agents. Presence, 7(6):564–587, 1998. [5] N. Bertschinger, E. Olbrich, N. Ay, and J. Jost. Autonomy: An information theoretic perspective. Biosystems, 91(2):331–345, 2008.

[6] A. Seth. Measuring autonomy and emergence via Granger causality. Artificial Life, 16(2):179–196, 2010.

[7] D. Vernon. Artificial Cognitive Systems — A Primer. MIT Press, Cambridge, MA, 2014.

[8] T. Ziemke. On the role of emotion in biological and robotic autonomy. BioSystems, 91:401– 408, 2008.

[9] T. B. Sheridan and W. L. Verplank. Human and computer control for undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory, 1978.

[10] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. Proceedings of the IEEE, 94(7), 2006.

[11] J. M. Bradshaw, P. J. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok. Dimensions of adjustable autonomy and mixed-initiative interaction. In M. Nickles, M. Rovatos, and G. Weiss, editors, Agents and Computational Autonomy: Potential, Risks, and Solutions, volume 2969 of LNAI, pages 17–39. Springer, Berlin/Heidelberg, 2004.

[12] http://journal.frontiersin.org/researchtopic/6534/shared-autonomy.

[13] X. Barandiaran and A. Moreno. Adaptivity: From metabolism to behavior. Adaptive Behavior, 16(5):325–344, 2008.

[14] H. R. Maturana and F. J. Varela. Autopoiesis and Cognition — The Realization of the Living. Boston Studies on the Philosophy of Science. D. Reidel Publishing Company, Dordrecht, Holland, 1980.

[15] M. H. Bickhard. Autonomy, function, and representation. Communication and Control— Artificial Intelligence, 17(3–4):111–131, 2000.

[16] W. D. Christensen and C. A. Hooker. An interactivist-constructivist approach to intelligence: self-directed anticipative learning. Philosophical Psychology, 13(1):5–45, 2000.

[17] J. O. Kephart and D. M. Chess. The vision of autonomic computing. IEEE Computer, 36(1):41–50, 2003.

[18] IBM. An architectural blueprint for autonomic computing. White paper, 2005.

[19] J. L. Crowley, D. Hall, and R. Emonet. Autonomic computer vision systems. In The 5th International Conference on Computer Vision Systems, 2007.

[20] K. R. Th́orisson and H. P. Helgasson. Cognitive architectures and autonomy: A compara- tive review. Journal of Artificial General Intelligence, 3(2):1–30, 2012.

[21] R. So and L. Sonenberg. Agents with initiative: A preliminary report. In M. Nickles, M. Rovatsos, and G. Weiss, editors, Autonomy 2003, LNAI 2969, pages 237–248. Springer- Verlag, Berlin Heidelberg, 2004.

[22] M. L. Anderson, S. Fults, D. P. Josyula, T. Oates, D. Perlis, M. Schmill, S. Wilson, and D. Wright. A self-help guide for autonomous systems. AI Magazine, 29(2), 2008.

[23] M. Vincze and D. Vernon. Industrial priorities for cognitive robotics. RockEU2 Deliverable D3.1, 2017.

[24] D. Vernon and M. Vincze. Industrial priorities for cognitive robotics. In R. Chrisley, V. C. Müller, Y. Sandamirskaya, and M. Vincze, editors, Proceedings of EUCognition 2016, Cognitive Robot Architectures, volume CEUR-WS Vol-1855, pages 6–9, Vienna, December 2017. European Society for Cognitive Systems.

[25] H. von Foerster. Understanding Understanding: Essays on Cybernetics and Cognition. Springer, New York, 2003.

[26] D. Vernon. Two ways (not) to design a cognitive architecture. In R. Chrisley, V. C. Müller, Y. Sandamirskaya, and M. Vincze, editors, Proceedings of EUCognition 2016, Cognitive Robot Architectures, volume CEUR-WS Vol-1855, pages 42–43, Vienna, December 2017. European Society for Cognitive Systems.

[27] R. Sun. Desiderata for cognitive architectures. Philosophical Psychology, 17(3):341–373, 2004.

[28] D. Vernon, C. von Hofsten, and L. Fadiga. Desiderata for developmental cognitive archttectures. Biologically Inspired Cognitive Architectures, 18:116–127, 2016.

Appendix I

The key goal of Work Package 3 is to accelerate the deployment of cognitive systems in industrial applications and to identify the technologies that most urgently need further development. It comprises five tasks:

Task 3.1: Industrial priorities (months 1-18)
Task 3.2: Catalogue of Cognitive Systems Capabilities (months 6-18)
Task 3.3: RockEU2 Cognitive Architecture Schema (months 1-24)
Task 3.4: Cognition-Autonomy Framework (months 1-18)
Task 3.5: Software Engineering Factors in Cognitive Robotics (months 9-21)

The original proposal identified some links between these tasks, stating that Task 3.1 would provide input for Task 3.3, 3. 4, and 3.5, and that Task 2 would provide input for Task 3.1, 3.3, 3.4, and 3.5. While indicative of their mutual relevance, these links lack the specificity necessary to render them operationally useful. Here, we aim to make explicit the flow of information between these five tasks and to highlight what each task will endeavour to do with that information.

Figure A1 depicts a flow-graph showing the five tasks and their outputs.[2] Some of these outputs are deliverables while others encapsulate information that will be derived from these deliverables.

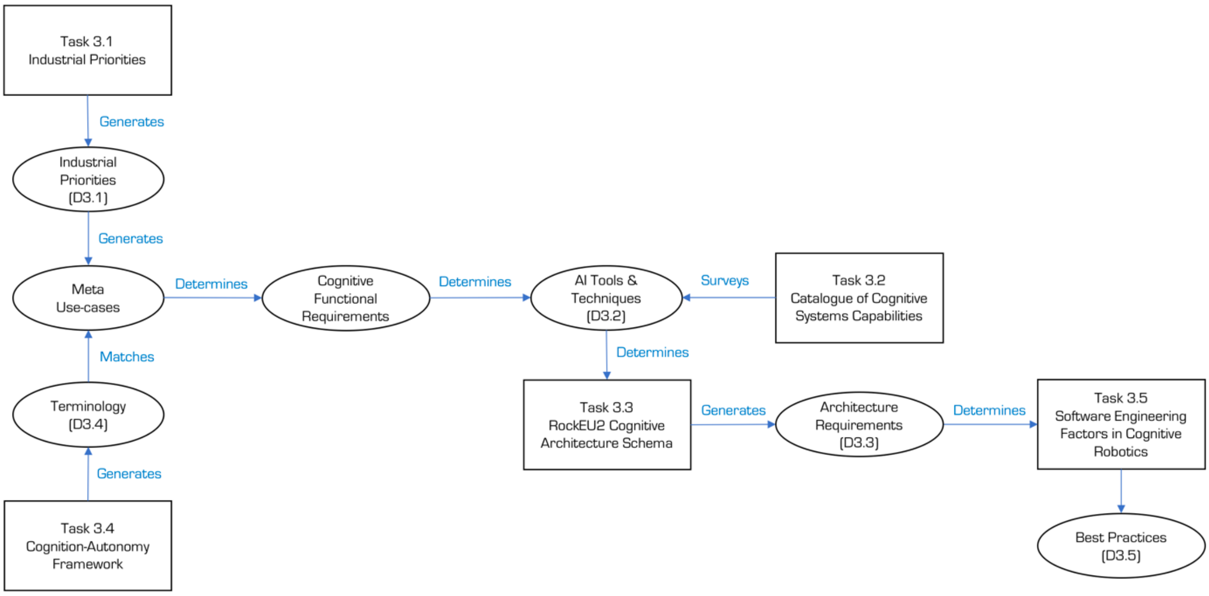

Figure A1: Information flow among the five task in Work Package 3 – Cognitive System Coordination.

Let's look at each of these outputs. The industrial priorities document, which is the product of Task 3.1, is substantially complete, although we intend to treat it as a living document and continually update it in the light of future inputs from industry. It comprises, essentially, eleven desiderata for cognitive robotics. These desiderata represent a distillation of the views of many industrial developers and, by extension, their customers. However, the distillation has rendered them somewhat abstract in formulation and they are difficult to map directly to the AI tools and techniques that are required to make them a reality, or even indirectly through an intermediate list of requirements for cognitive functionality. Therefore, we plan in the

[2] Although Figure 1 shows the information flow among the work package 3 tasks, it should not be interpreted as a schedule of tasks (e.g. as a form of PERT chart). All tasks are operational concurrently.

short-term to expose the essence of all eleven desiderata is a meta[3] use-case. This will be used to provide the list of requirements for cognitive functionality, which in turn will determine the requisite AI tools and techniques.

Notably, this use-case uses the terminology that results from an exercise (Task 3.4) in viewing cognitive robots as autonomous systems and highlighting the language required to specify autonomous behaviour as opposed to the language used to describe cognitive behaviour.

A parallel task, Task 3.2, to create a catalogue of implemented and accessible cognitive systems capabilities has the goal of identifying the AI tools and techniques that can be deployed today to satisfy the cognitive functional requirements. Ideally the two lists of AI techniques and tools – one resulting from Tasks 3.1 and 3.4 and one from Task 3.2 – will be identical. Where they are not, the differences indicate the work that needs to be done to ensure the deployment of cognitive robotics in industry.

It is one thing – albeit a crucial and indispensable one – to have a catalogue of AI tools and techniques, but it another to be able to incorporate them in a complete operational system. This non-trivial integration exercise is typical effected with the aid of a system architecture, or a cognitive system architecture in the case of a cognitive robot. Since architectures tend to be application-dependent, Task 3.3 endeavours to map the catalogue of AI tools and techniques to cognitive architecture requirements, also known as a cognitive architecture schema.

In turn, these architecture requirements must be converted to a fully functional system. That requires sophisticated software engineering. While there is more than one popular software engineering paradigm in robotics – component-based software engineering and model-based software engineering to name two of the most prominent – the deployment of that paradigm should factor in the type of software architecture that is required to integrate the AI tools and techniques into an effective cognitive robot control system. That is what we aim to do in Task 3.5.

[3] They are meta use-cases because they are more generic in their description than would be a use-case tied to a real target application.