INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018
https://doi.org/10.21278/idc.2018.0281

FRAMING THE CONCEPT OF AUTONOMY IN SYSTEM DESIGN

T. F. Beernaert, A. E. Bayrak, L. F. P. Etman and P. Y. Papalambros

Abstract

Advancements in multiple domains have led to a focus on autonomy in engineered systems. This focus necessitates a clear definition of the term ‘autonomy’ and its implications for system design. To this end, we present a framework that defines autonomy mathematically as a function of independence and task complexity. A deeper understanding is established by generating design principles that can be used to influence autonomy. We illustrate these concepts on autonomy quantification for a coffee machine and on a qualitative retrospective assessment of the evolution of autonomy for the automobile.

Keywords: autonomy, independence, complexity, characteristics and properties, design guidelines

1. Introduction

Autonomy of engineered systems is a major current technological trend. For example, the development and assimilation of autonomous vehicles in transportation systems is expected to have major social and economic impact. Similarly, increasingly autonomous everyday systems, such as household devices, and their connection through the internet of things may drastically affect the way we live. While the term 'autonomy' is widely used, articulating its precise definition and ways to measure the degree of 'autonomy' of a particular system appears incomplete in the literature.

For example, washing machines intuitively appear to have a low degree of autonomy, while Waymo's self-driving vehicles, to be available to the public in the near future (Wakabayashi, 2017), appear to be highly autonomous; the iRobot Roomba (iRobot, 2017) is a self-navigating vacuum cleaner that requires no human operator during cleaning, and its autonomy might be placed between that of the washing machine and the self-driving car. We hypothesize that defining and measuring autonomy in a more precise way will support design analysis of a desired degree of autonomy and ways to increase or decrease a system’s autonomy for a particular set of tasks.

This paper presents a framework for defining and calculating the degree of autonomy in the design of a system. While there is still no clear consensus on the definition of autonomy, we identify important properties based on the existing literature. We define autonomy as a mode of operation that can be evaluated along a half-closed, bounded, and continuous interval (i.e., it can be a value in $[0, 1)$) and that has meaning only when both the system and the task it is executing are considered; moreover, we argue that its assessment can change over time.

We decompose system autonomy into system independence and task complexity, that can be individually measured. Independence refers to the system's engagement in the execution of a task. Task complexity refers to the diversity in decision making under various options and uncertainties. From the proposed autonomy framework, we extract two design principles that can be used to vary a system's autonomy, usually a desired increased autonomy: (1) remove the interactions from humans and other systems in the environment and (2) enable the system to make a wider range of choices.

We demonstrate the use of this conceptual framework on the design of a more autonomous coffee machine and in assessing autonomy of an automobile. The coffee machine example shows how the method can be used to calculate and increase autonomy, and the automobile assessment qualitatively shows the effect of past and current developments in this domain of autonomy.

In the remainder of the paper, we review the existing literature to identify research gaps and form a basis for the framework in Section 2. In Section 3, the framework is presented and formulated, followed by an elaboration of the properties independence and task complexity. We deduce a set of design principles from the framework in Section 4 and demonstrate its use and contributions in Section 5.

2. Literature review

In this section, we review the existing frameworks for defining and modeling autonomy of engineering systems, and set the stage for further elucidation.

2.1. Existing frameworks

One recurring method in literature for the classification of autonomy is through a chart of levels. Every level is accompanied by a textual description of criteria for the system and its surroundings. Some of these frameworks feature abstract descriptions, making them suitable for a wide range of applications (Endsley and Kaber, 1999; Parasuraman et al., 2000; Huang et al., 2007; Lodwich, 2016), while others are tailored for a specific domain to decrease ambiguity. Examples of the last category can be found in autonomy evaluations for UAV's (Unmanned Aerial Vehicles) (Clough, 2002) or motorized vehicles (SAE, 2016). Though being easily understandable, this type of classification is largely subjective, because it lacks mathematical foundations and depends on human interpretation.

The works of Sheridan et al. (1978) and Parasuraman et al. (2000), later adapted by Beer et al. (2014), extend an abstract categorization by a set of guidelines on how to determine autonomy and its effects on the system's surroundings. Their results provide a thorough, clear understanding of the symbiosis between a system and its (human) operator.

Although many researchers acknowledge that autonomy can be represented on a spectrum between 0 and 1, few define it as a continuous quantity. Yanco and Drury (2004) cover many aspects that should determine a scale for system's autonomy, but evaluate systems subjectively on this scale. The work of Barber et al. (2000) considers multi-agent systems and assesses each individual agent's autonomy as its relative decision strength in reaching a common goal. Finally, Hrabia et al. (2015) propose various system capabilities as metrics to calculate the degree of autonomy. The works in this paragraph provide methods on how to quantify autonomy, yet their definitions do not match our present approach introduced in Section 2.3.

2.2. Framework properties

Review of the literature indicate that existing methods to evaluate autonomy are often based on qualitative assessments, are tailored to specific domains, and do not provide relations between autonomy of a system and design choices. Therefore, we focus on the following elements:

1. ***Quantification:*** The desired framework should have a mathematical representation, enabling rigor and integration, thus allowing domain application and consistency in results. The framework should also build upon system characteristics the designer can influence, to be able to trace the implications of design choices on autonomy.
2. ***Generality:*** While many domain-specific methods to assess autonomy are available. The present effort aims at a high level of abstraction to make it applicable to a wide range of domains.
3. ***Design synthesis:*** Developing a black-box function that computes autonomy is appealing per se, but such a method becomes more valuable and useful when it enable investigation into how the system could be altered to increase autonomy. We therefore seek to develop a clearer understanding on how a system can evolve towards a more autonomous state.

2.3. Autonomy definitions and properties

In this section, we aim to distil key definitions and properties in line with most of the literature. The findings from the literature form the foundation of the new definition and framework. In this effort we assert that autonomy is not just a measure of a system's capability but of a measure of a system's behavior in a task context, namely, it is a function of the system's independence and task complexity. Although task complexity remains constant, tasks become relatively easier due to technological advancement.

Autonomy can be expressed in a single metric that is quantifiable on a continuum between 0% (no autonomy) and 100% (full autonomy) (Endsley and Kaber, 1999; Parasuraman et al., 2000; Huang et al., 2007). Autonomy is a property of a system's mode of operation, and therefore only has a meaning within a certain task context (Beer et al., 2014). For example, Waymo's self-driving car may have a different value of autonomy for the tasks 'driving' and 'parking'. Instead of saying 'this system is autonomous', we should say 'this system operates autonomously'. Autonomy is a function of both system and task. This presumption extends the quantification method of Hrabia et al. (2015).

Another characteristic of autonomy extracted from the literature is a high degree of independence (Antsaklis and Passino, 1989; Wooldridge and Jennings, 1995; Barber et al., 2000; Murphy, 2000; Bradshaw et al., 2004; Schillo and Fischer, 2004; Yanco and Drury, 2004; Bekey, 2005; Huang et al., 2007; Thórisson and Helgasson, 2012; Beer et al., 2014). Following these studies, we define independence as follows:

Definition 1: A system's independence is its degree of initiative in the execution of a task, relative to that of external entities.

A highly independent system initiates most of the interactions necessary to execute a task, whereas a less independent system relies on more input from other entities. However, independence cannot be used as the sole characteristic of autonomy. Though a washing machine, the Roomba and a self-driving car would all be considered quite independent, one could use the term ‘automatic’ instead of autonomous to further distinguish them. The difference between autonomy and automation, discussed in several research papers (Clough, 2002; Thórisson and Helgasson, 2012; Hrabia et al., 2015; Lodwich, 2016), is important. It is not possible to distinguish autonomy from automation using only independence. The washing machine could be considered a fairly independent system, but we argue that this system is automatic as opposed to autonomous, because the task it accomplishes is relatively ‘simple’. The Roomba and the self-driving car might be valued similarly independent as the washing machine, but why is it that we are more likely to use the term ‘autonomous’ for these systems? Clough (2002) states that the difference is the presence of “free will.” We can accurately predict the behavior of a washing machine going through its predetermined washing cycles, but associate the movements and orientation of the Roomba as the device having ‘a mind of its own’. The notion of ‘free will’ here is ambiguous and subjective. A young child might be fascinated by the ‘magic’ that causes the same washing machine to rinse and spin on its own, while the engineer that designed the control algorithm of the vacuum robot might simply think that its movements are all pre-programmed and therefore predictable.

Here we set such aside a philosophical discussion about free will and concentrate on a more readily quantifiable model. According to Hrabia et al. (2015), automated systems operate in a well-defined, constant environment and they always respond to the same input with the same output. In line with Castelfranchi and Falcone (2003), we consider automation to be a subcategory of autonomy. We conclude from their work that the difference originates from the system property ‘intelligence’ and the task property ‘complexity’. Intelligence is a property that is closely related to autonomy by some researchers (Zeigler, 1990; Huang et al., 2007; Thórisson and Helgasson, 2012; Beer et al., 2014). However, we agree with Clough (2002) that this should not be a necessary component of autonomy; according to his example, a bacterium (low intelligence) is more autonomous than a toddler (high intelligence). Instead, we view intelligence as a tool that can be used to increase autonomy and/or performance.

We use the term 'task complexity' to distinguish automation from autonomy and follow Campbell (1988), who based his framework on task characteristics that increase information load, information diversity and rate of information change; he identified four sources of complexity: (1) multiple paths to a desired end-state, (2) multiple desired end-states, (3) conflicting interdependence and (4) uncertainty or probabilistic linkages. Here, task complexity reflects environmental uncertainty, unpredictability and decision-making complexity, and even hints 'free will' or 'freedom' in a single property. This concept covers aspects of existing works regarding autonomy, such as the presence of significant uncertainties (Sheridan et al., 1978; Antsaklis and Passino, 1989), unforeseen variations (Thórisson and Helgasson, 2012), environmental changes (Zeigler, 1990; Murphy, 2000; Thrun, 2004) and a real-world environment (Bekey, 2005). It extends the quantified frameworks of Barber et al. (2000) and Hrabia et al. (2015), which only consider system capabilities. In a fully determined environment, we can define task complexity as follows:

Definition 2: Task complexity is the number of possible ways in which a task can be executed.

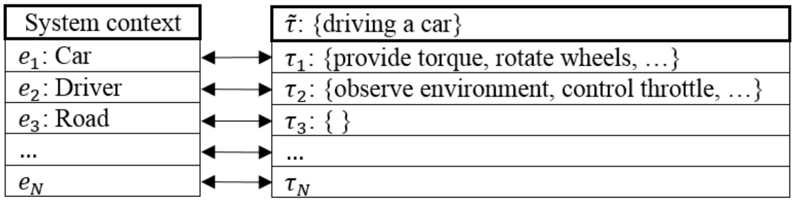
Note, however, that this definition only reflects the deterministic part of our concept of task complexity, and disregards uncertainty and environmental changes. The perception of task complexity, and therefore autonomy, is time-dependent. A task considered to be complex today might become trivial in ten years. The evaluation of task complexity itself will not change, but the scale on which we assess it will expand continuously. The first independent washing machine would have been more autonomous at the time, but has shifted into automation as the standards of task complexity have increased. Since we expect this trend to continue in the foreseeable future, we presume that autonomy of a constant system with a constant task will decay over time.

3. Method

We outline the foundation of the proposed framework for autonomy: the proposed concept of autonomy with the accompanying mathematical formulation.

3.1. Assumptions

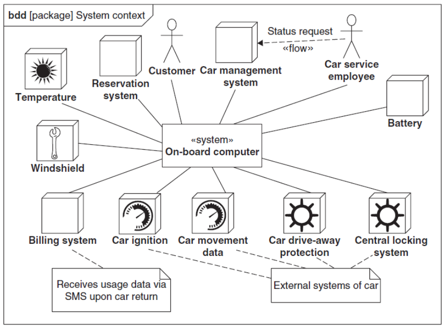
We assume that a system has the purpose of fulfilling a high-level task $\tilde{\tau}$ within a context of external entities (humans and systems), which contribute to and benefit from the execution of these tasks. The system context is a network of N entities e_i $(i = 1,2, \dots, N)$, where e_1 is the system itself and e_2 through e_N are the externals. The high-level task $\tilde{\tau}$ is decomposed, such that τ_i contains all subtasks that are executed by e_i. τ_i is the largest set of tasks that can be assigned to a single entity e_i and may be empty. A task τ_i is a general description of what an entity does. How entities execute their task can be influenced by the designer and will partly determine the corresponding autonomy. Tasks relate to the desired behavior in time, which excludes static functions such as 'storing liquid'. Figure 1 shows an example of the high-level task 'driving a car' and how it is decomposed and related to the system context.



System context		$\tilde{\tau}$: {driving a car}
e_1: Car	↔	τ_1: {provide torque, rotate wheels, …}
e_2: Driver	↔	τ_2: {observe environment, control throttle, …}
e_3: Road	↔	τ_3: { }
...	↔	...
e_N	↔	τ_N

Figure 1. Mapping between entities and tasks for the high-level task of driving a car

In the model representation, we include interactions between all entities necessary to conduct $\tilde{\tau}$ in the system context network. The establishment of this network implies the definition of the system boundary (i.e., entities included as part of the system context), which is crucial for the assessment of autonomy. Lodwich (2016) illustrates that a minor modification of the system boundary can drastically change our perception of autonomy. In multi-agent systems, autonomy can be assessed for a single agent but, by expanding the context, also for the complete set of agents. Alternatively, we can focus on one system and investigate the autonomy of its submodules. Figure 2 shows an example of a system context diagram, modelled in SysML (Weilkiens, 2007). The directed adjacency matrix representation of the system context is deemed more suitable for our work, since it is more compact and easier to incorporate in a mathematical formulation.



$$
\begin{array}{c|ccccc|}
 & \boldsymbol{e_1} & \boldsymbol{e_2} & \boldsymbol{e_3} & \cdots & \boldsymbol{e_{N_e}} \\
\hline
\boldsymbol{e_1} & \blacksquare & 1 & & & \\
\boldsymbol{e_2} & 1 & \blacksquare & 1 & & 1 \\
\boldsymbol{e_3} & & & \blacksquare & & 1 \\
\vdots & & & & \ddots & \\
\boldsymbol{e_{N_e}} & & & 1 & & \blacksquare \\
\hline
\end{array}
$$

Figure 2. Examples of a system context network (left) (Weilkiens, 2007) and an adjacency matrix (right)

Finally, we assume that the systems that are evaluated always have a satisfactory (or feasible, in optimization terminology) performance, i.e. result in outcomes that meet the designer's and/or stakeholder's requirements This is an important assumption that simplifies modelling.

3.2. Proposed autonomy definition

Following our literature review and assumptions, we define autonomy as follows:

Definition 3: Autonomy is the mode of operation in which a system executes a complex task, independently from humans and external systems.

A system that operates autonomously makes decisions regarding a complex task, and carries them out by itself. This definition comes closest to the ALFUS framework presented by Huang et al. (2007), where the concept of task complexity is further decomposed into mission complexity and environmental difficulty. To avoid misinterpretation, it is important to consider the definitions of task complexity and independence as we use them here. Also, note that we do not link autonomy to the performance of the system. According to our definition it will be easy to make any system highly autonomous, simply by giving it vast decision freedom and completely removing all interactions from external entities. The majority of systems, however, may fail to have a satisfactory performance because they were not designed to operate in such extreme circumstances.

Various works discussed in Parasuraman et al. (2000) observe no correlation between autonomy and performance. In some cases, an increase in autonomy will decrease performance (negative correlation); The previously mentioned Roomba, for instance, may take more time to clean a room than a classical vacuum cleaner manually operated by a human. Also, the cleanliness of the room may be lower, because the human uses visual feedback mechanism to check whether the room is sufficiently clean. Performance can also be considered in the context of Waymo's self-driving car. According to Urmson (2015), chances of accidents significantly drop in comparison to conventional vehicles, implying a positive correlation between system autonomy and super-system performance (safety).

3.3. Proposed framework

We locate autonomy in the space of independence and task complexity, similar to the automation versus task entropy grid in Sheridan et al. (1978). The framework uses independence instead of automation because, as previously elaborated, we consider this to be the underlying property that determines autonomy.

We evaluate independence on the interval $[0,1]$ and define it as a function of the system's behavior in a specific task setting. Task complexity is a function of the system's decision space for a task, and is valued on a scale $[0, \infty)$. If we imagine a task, we can always create another task that requires more decision effort, therefore this dimension is not bounded from above.

We would like autonomy to be bounded in its properties, i.e. autonomy should take on finite values for the complete range of independence and task complexity. This means that autonomy can be proportional to independence, but has to be asymptotic with respect to task complexity. The use of an exponential function is motivated by task complexity's lower boundary; zero task complexity should also lead to zero autonomy. Given these assumptions, we express autonomy α_i of entity e_i, executing task τ_i, as:

$$\alpha_i = I_i\left(1 - e^{-\frac{C_i}{\lambda}}\right) \tag{1}$$

where I_i is the independence of e_i, C_i is the task complexity of τ_i and λ is a benchmark task complexity which is further elaborated in Section 3.5. This function gives zero autonomy for the lower bounds of I_i and C_i. Full autonomy $(\alpha_i = 1)$ can only be achieved if the entity is completely independent $(I_i = 1)$ and makes infinitely complex decisions $(C_i \to \infty)$. We explicitly state that full autonomy is utopic and can never be attained, due to asymptotic behavior in task complexity. Systems with zero autonomy do exist in the form of 'passive' systems, i.e. incapable of making decisions, such as a hammer. Therefore, autonomy takes values on the interval $[0, 1)$. This formulation implies that an operating system cannot have a high degree of autonomy if it has only high independence or high task complexity, since we require the combination of both. The autonomy function is monotonically increasing in I_i and C_i, and monotonically decreasing in λ; any design optimization formulation involving autonomy will need to include additional objectives and constraints to be meaningful.

Figure 3 shows the two-dimensional independence-complexity space wherein systems can be located. Now we can further distinguish automation from autonomy. Full automation refers to a system that operates completely independently but always executes its task in the same way regardless of environmental uncertainties. For example, imagine a machine that labels plastic bottles without any external influence. If this machine operates with no other input, we would consider it as a fully automated system. This machine could become more autonomous (and less automated) if it had the ability to sort out bottles with misplaced labels.

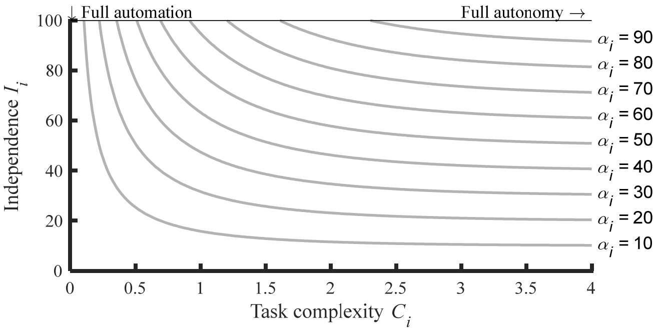


Figure 3. Independence-complexity space with constant autonomy lines ($\boldsymbol{\lambda = 1}$)

A weighted sum can be used to calculate the autonomy of entity i for multiple high-level tasks $\tilde{\tau}_j$ $(j = 1,2, \dots, N_\tau)$. Normalized task weights $w^{(j)}$ can be assigned subjectively, but computing them from task characteristics (e.g., frequency, time, criticality) would offer a more consistent approach. These weights could also be functions of $\alpha_1^{(j)}$, giving higher weights to tasks that correspond to low system autonomy. In any case, the autonomy of e_i for multiple tasks $\tilde{\tau}_j$ is computed as:

$$\alpha_i^{(\tau)} = \sum_{j=1}^{N_\tau} w^{(j)} \alpha_i^{(j)} \tag{2}$$

To investigate an entity's (human or system) autonomy, an abstract representation of its behavior in the environment is necessary. Such models have been developed and used in the domains of robotics (Zeigler, 1990; Huang et al., 2007), cognitive psychology (Broadbent, 1958) and military combat (Bazin, 2005). From a system's perspective, the behavior of a system can be represented as a combination of input, internal operation, and resulting output. Any task can be decomposed into a combination of these three elements. This simple functional representation appears sufficient for our purpose. Inputs and outputs describe how an entity interacts with its environment and directly determine its independence. Decisions made in the internal operation are linked to the notion of task complexity, as introduced earlier. The following two sections propose mathematical formulations for independence and task complexity.

3.4. Independence

Using an information-processing model, we can create a network representation where nodes are entities and edges are interactions. An entity B (Figure 4) can send information to entity C that causes a behavior change in C; or it can solicit information from entity A that causes no behavior change in A. *Actions* and *perceptions* are the two forms of interaction between two entities:

Definition 4: Action is information from a sender entity to a receiver entity that causes behavior change in the receiver.

Definition 5: Perception or sensing is information requested from a sender to a receiver about the receiver that causes no behavior change in the receiver.

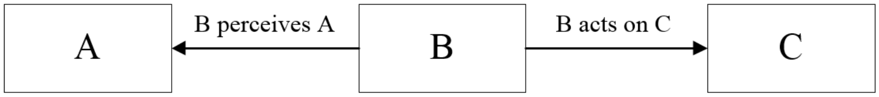
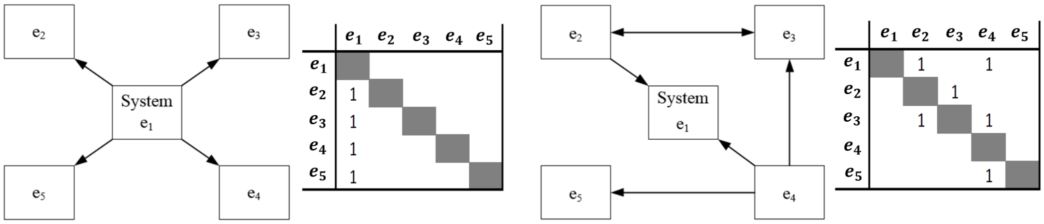


Figure 4. Definition of directed cues in proposed interaction network

Such a network represents the system designer's solution to accomplish a task and therefore depends on the realisation (embodiment) of the system. The system designer chooses which interactions are being used and where decisions are made, thereby changing independence (and indirectly influencing autonomy). Maximum system independence occurs when the system initiates every sensing and action in the network. There are no interactions initiated by any of the external entities and they only experience the actions from the system. Figure 5 (left) displays this extreme case with 100% independence, where all interactions are initiated by the system. The same figure (right) shows the other extreme, where the system initiates no interaction (0% independence). In the context of independence, there is no need to distinguish actions from perceptions, so hereon we will use only the term interaction.



	e_1	e_2	e_3	e_4	e_5
e_1	■				
e_2	1	■			
e_3	1		■		
e_4	1			■	
e_5	1				■

	e_1	e_2	e_3	e_4	e_5
e_1	■	1		1	
e_2		■	1		
e_3		1	■	1	
e_4				■	
e_5				1	■

Figure 5. Interaction networks and adjacency matrices with full system independence (left) and no system independence (right)

To calculate the independence of a system with multiple entities, let A be the adjacency matrix of the interaction network that corresponds to $\tilde{\tau}$. We then define the independence of e_i as the sum of its initiated interactions, normalized by all interactions in the network:

$$I_i = \frac{\sum_{k=1}^{N} A_{ik}}{\sum_{k=1}^{N} \sum_{i'=1}^{N} A_{i'k}} \tag{3}$$

where i and k refer to the sending and receiving entity, respectively. Note that the sum of all entities' independence for $\tilde{\tau}$ is equal to one. The binary values in the adjacency matrix can be replaced by real numbers to reflect relative interaction strength. This strength value can be based on interaction frequency, time, criticality, priority etc., with the formulation remaining the same. When the interaction strength is a measureable quantity, then the value of an entity's independence can be validated by an experiment.

3.5. Task complexity

At present we assume that task complexity can be quantified in a single dimension. We stated in Section 3.3 that task complexity can be infinitely large. To prevent ill-conditioning of Equation (1) for large values of C_i, a scaling factor λ is introduced. This factor is called the benchmark task complexity and represents the status quo. The factor λ is a measure for the most complex task we can achieve at any point in time. Therefore, λ depends on technical, economic and environmental factors. We have no explicit functional form for this quantity at present and we assume it as given.

From Equation (1), since e^{-x} becomes very small for $x \geq 5$, the practical range of task complexity for autonomy is $[0, 5\lambda]$, with tasks above 5λ having approximately the same autonomy for constant I. The value of 5λ is not a limit; rather, it reflects the continuously changing interpretation of a high complexity task. Any function for computing task complexity should have a range in the above interval.

Haerem et al. (2015) proposed the number of ways to execute a task as a metric for complexity. Their method requires an a-priori generated network of actions and can therefore formalize the deterministic part of our current concept of task complexity. However it is unable to incorporate uncertainty within the task itself. A suitable metric for uncertainty has not yet been found and is necessary to be incorporated for a complete quantification of task complexity.

A very simple and, unfortunately, subjective method to evaluate task complexity C_i is by qualitatively assigning a value relative to 5λ, based on the four sources in Section 2.3. We stress again that any method, even if quantified, requires an accurate description of system and task. A vacuum cleaner might clean a room with less task complexity than a human operator. Though the entities' tasks remain the same and are graded with a constant task complexity, they become relatively less complex as λ increases. In our statement that tasks maintain their complexity over time, we imply that the context remains exactly the same. The task of driving a car, for instance, has increased in absolute task complexity over the last decades, because of an increasing amount of vehicle features (hence, interactions) and a more complex infrastructure. 'Driving a car in 1960' and 'driving a car in 2017' are two different tasks and, therefore, have different complexities. Since we evaluate task complexity on a λ-dependent scale, the actual value of λ is unimportant and we assume $\lambda = 1$ in our analyses.

As an example, assume we qualitatively evaluate a system's task to be of 'medium' complexity and refer to 'medium' as 50% of 5λ. We then acquire a value of $C_1 = 2.5$ for $\lambda = 1$, and use this to determine the system's autonomy as $\alpha_1 = 0.46$ (where $I_1 = 0.5$). If we assess autonomy of this system for a changing technological standard, we can observe a decay as the benchmark task complexity increases, as shown in Figure 6.

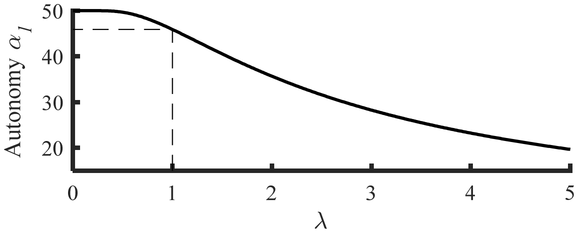


Figure 6. Autonomy decays over technological advancement

If the same situation is evaluated in a more technologically advanced future where λ has doubled, it should be considered proportionally easier than before. The task complexity will have dropped to 25% of 5λ, remaining at the previously acquired value $C_1 = 2.5$. With this example, we want to show that even though a task maintains a constant absolute task complexity, it becomes relatively easier as λ increases.

4. Design principles to increase autonomy

Using the introduced framework, we extract two principles that a designer can use to increase the system's autonomy. The first proposes alterations to the system implementation and relates to independence, the second affects the task scope and hence, task complexity. These principles assume that the design decisions are made without sacrificing system performance, as noted in Section 3.1.

Design Principle 1: To increase autonomy, remove interactions from external entities by introducing new interactions, initiated by the system.

The designer has to remove interactions that are initiated by external entities, see Equation (3). To maintain functionality and performance regarding the task, interactions from the system can be added. An example of this principle is the evolution of a lighting system in a room. The user flipping a switch (action from external to system) is replaced by a motion sensor that detects a person in the room (perception from system to external), triggering the on/off state of the light. By the definitions introduced in this work, the motion-actuated lighting system is more autonomous than the manual lighting system.

Design Principle 2: To increase autonomy, increase the task complexity of the entity's task.

Autonomy will increase when the system has the authority to make more diverse choices. If the same lighting system can be upgraded to adjust brightness based on the time of the day or the weather, the system autonomy increases.

These two principles can be applied for each individual task, or a designer might focus only on tasks with larger weights in Equation (2).

5. Illustrative examples

We present some simple examples to demonstrate how the above ideas can be sued.

5.1. Coffee machine

Consider a capsule-based coffee machine's autonomy for brewing coffee. First, the boundary is defined and the external entities identified. SysML's sequence diagram (Weilkiens, 2007) is a suitable source of information for the interaction network. We assign each interaction a value based on how long it occurs in seconds. Interactions that occur more than once are simply superimposed, and their values are summed. Using the introduced equations, we find an independence value for every entity. Note that entities that do not initiate any interaction are completely dependent (zero independence). As stated before, task complexity is evaluated subjectively and requires a more profound method to enable rigor. For now, we assume that the coffee machine executes its tasks in a relatively simple fashion and assign a value of $C_1 = 0.5$. The user makes more choices and is awarded $C_2 = 1$.

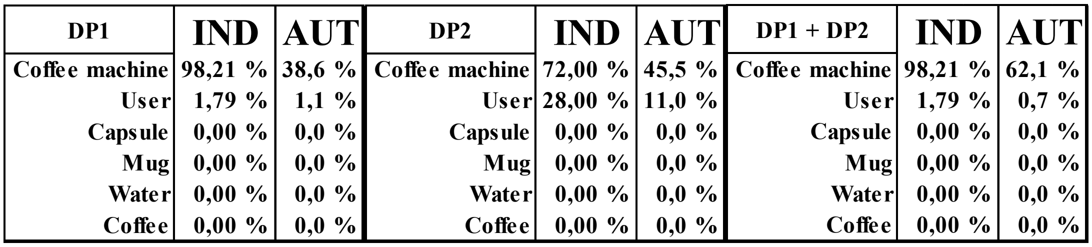
	Time [s]
Take capsule	5
Place capsule	10
Take cup	15
Place cup	3
Turn on	1
Select concentration	1
Heat	15
Brew coffee	15
Dispense coffee	15

Coffee machine	▨	15				
User		▨				
Capsule	15	5	▨			
Cup	15	15		▨		
Water	30				▨	
Coffee	30					▨

	IND	AUT
Coffee machine	72,00 %	28,3 %
User	28,00 %	17,7 %
Capsule	0,00 %	0,0 %
Mug	0,00 %	0,0 %
Water	0,00 %	0,0 %
Coffee	0,00 %	0,0 %

Figure 7. Timed interactions (left), interaction adjacency matrix (centre) and resulting independence and autonomy degrees per entity (right)

We apply design principle 1 and identify that the interactions from the user should be removed. It might be infeasible or unprofitable, but enabling the machine to take capsules and cups by itself will increase its autonomy. Design principle 2 advises to increase the diversity of choices the machine can make. The system's autonomy will increase when it is able to detect and observe the user and select the best suitable type of coffee and concentration based on whatever data it acquired. Figure 8 shows the resulting independence and autonomy after applying the design principles. Principle 1 is demonstrated by rerouting the interactions 'take capsule' and 'take cup' from coffee machine to capsule and cup, respectively, and removing the interactions 'place capsule' and 'place cup'. Principle 2 is realised by swapping the task complexity of the system and human, i.e. $C_1 = 1$ and $C_2 = 0.5$.



DP1	IND	AUT	DP2	IND	AUT	DP1 + DP2	IND	AUT
Coffee machine	98,21 %	38,6 %	Coffee machine	72,00 %	45,5 %	Coffee machine	98,21 %	62,1 %
User	1,79 %	1,1 %	User	28,00 %	11,0 %	User	1,79 %	0,7 %
Capsule	0,00 %	0,0 %	Capsule	0,00 %	0,0 %	Capsule	0,00 %	0,0 %
Mug	0,00 %	0,0 %	Mug	0,00 %	0,0 %	Mug	0,00 %	0,0 %
Water	0,00 %	0,0 %	Water	0,00 %	0,0 %	Water	0,00 %	0,0 %
Coffee	0,00 %	0,0 %	Coffee	0,00 %	0,0 %	Coffee	0,00 %	0,0 %

Figure 8. Application of principle 1 (left), principle 2 (centre) and both (right)

We observe that both design principles lead to an increase in system autonomy and reduction in user autonomy, and show synergy when combined. The application of the principles together increases system autonomy more than the sum of the separate gains from each principle. While we do not propose detailed design solutions in this example, we identify design targets to increase autonomy without sacrificing system performance when making detailed decisions.

5.2. Automobile evolution

Autonomy in automobiles has recently gained popularity. In a simple retrospective analysis, we assess past and current developments in automobiles within the proposed framework. This example contains a qualitative evaluation of independence and task complexity for the driver and the vehicle. The purpose is to show the historical trends in our framework while leaving the exact calculations to a more focused study on autonomous vehicle design. We consider the evolution of the automobile from its first appearance in the early 1900's, like Ford's Model T, through the developments of an electric starter, cruise control and an automatic transmission, to the current implementations of connected and smart cars. Figure 9 shows an indicative progression of the automobile and driver in the autonomy space for the task of driving a vehicle.

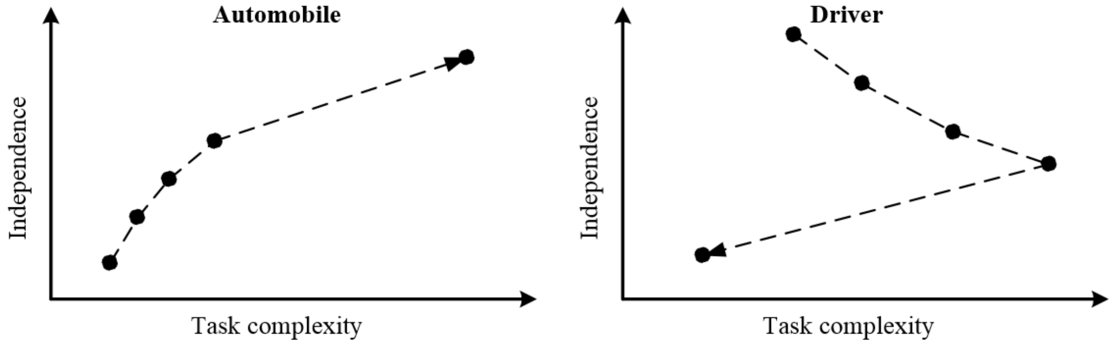


Figure 9. Qualitative assessment of past and current automobile evolutions in the independence-complexity space

Early evolutions have been automating certain processes in the vehicle, increasing the independence of the automobile and consequently decreasing that of the driver. As a result, the automobile's task complexity increased. However, because of an increasing number of vehicles and more complex infrastructure, the driver's task complexity has also grown. It is with current developments of smart, connected cars that we observe a major leap in autonomy. Taking the control away from the driver significantly reduces their independence and task complexity, while it drastically increases both properties in the automobile, and displays a gain in vehicle autonomy and reduction in driver autonomy. This assessment suggests that recent efforts to increase vehicle autonomy are motivated by the desire to reduce the driver's autonomy. This desire could be the underlying stimulus for all developments regarding autonomy, and provides an interesting perspective on technology development.

6. Conclusion

We have proposed a new generalized framework intended as a guide for designers in the development of autonomy in systems. The definition of autonomy is relative to the defined system boundary, and so the method can be used to assess autonomy in multi-agent systems or in submodules of a system. The mathematical formulation links autonomy to two properties, namely independence and task complexity. We consider a decay in task complexity, and therefore in autonomy, over time. The proposed method breaks these properties down to system behavior, and two design principles give advice on how to increase autonomy. Independence is calculated by an interaction network, but the current evaluation of task complexity is qualitative and therefore subjective. A more rigorous assessment or model of task complexity should be developed in a future study. The framework is demonstrated in a quantitative and a qualitative example for a coffee machine and automobile evolution, respectively. The coffee machine example shows how autonomy can be calculated, based on its behavior, and how it can be increased following the design principles. The qualitative application of the framework to past and current developments in the automotive domain shows a distinct progression and presumes that the implementation of smart, connected vehicles leads to a large increase in this system's autonomy. Finally we observe that the drive for technology for autonomy comes from a desire to decrease user autonomy rather than to increase system autonomy per se.

References

Antsaklis, P. and Passino, K. (1989), "Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues", *Journal of Intelligent and Robotic Systems*, Vol. 1 No. 4, pp. 315-342. https://doi.org/10.1007/BF00126465

Barber, K.S., Goel, A. and Martin, C.E. (2000), "Dynamic adaptive autonomy in multi-agent systems", *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 12, No. 2, pp. 129-147. https://doi.org/10.1080/095281300409793

Bazin, A. A. (2005), "Boyd's O-O-D-A Loop and the Infantry Company Commander", *Infantry Magazine*, Vol. 94 No. 1, pp. 17-19.

Beer, J. M., Fisk, A.D. and Rogers, W.A. (2014), "Toward a framework for levels of robot autonomy in human-robot interaction", *Journal of Human-Robot Interaction*, Vol. 3, No. 2, pp. 74-99. https://doi.org/10.5898/JHRI.3.2.Beer

Bekey, G. A. (2005), *Autonomous Robots; From Biological Inspiration to Implementation and Control*, The MIT Press, Cambridge

Bradshaw, J.M., Feltovich, P.J., Jung H., Kulkarni S., Taysom W. and Uszok A. (2004), "Dimensions of Adjustable Autonomy and Mixed-Initiative Interaction", *Agents and Computational Autonomy. AUTONOMY 2003. Lecture Notes in Computer Science, Vol. 2969*, pp. 17-39. https://doi.org/10.1007/978-3-540-25928-2_3

Broadbent, D.E. (1958), *Perception and Communication*, Pergamon, London, https://doi.org/10.1037/10037-000

Campbell, D.J. (1988), "Task Complexity: A Review and Analysis", *Academy of Management Review*, Vol. 13 No. 1, pp. 40-52. https://doi.org/10.5465/amr.1988.4306775

Castelfranchi, C. and Falcone, R. (2003), "From Automaticity to Autonomy: The Frontier of Artificial Agents", In: Hexmoor, H., Castelfranchi, C. and Falcone, R. (Eds.), *Agent Autonomy. Multiagent Systems, Artificial Societies, and Simulated Organizations (International Book Series)*, Vol 7., Springer, Boston, MA pp. 103-136. https://doi.org/10.1007/978-1-4419-9198-0_6

Clough, B.T. (2002), "Metrics, Schmetrics! How the heck do you determine a UAV's autonomy anyway?", *1st UAV Conference, Porsmouth, Virginia, May 20-23, 2002.* https://doi.org/10.2514/6.2002-3499

Endsley, M.R. and Kaber, D.B. (1999) "Level of automation effects on performance, situation awareness and workload in a dynamic control task", *Ergonomics*, Vol. 42 No. 3, pp. 462-492. https://doi.org/10.1080/001401399185595

Haerem, T., Pentland, B.T. and Miller, K.D. (2015), "Task Complexity: Extending a Core Concept", *Academy of Management Review*, Vol. 40, No. 3, pp. 446-460. https://doi.org/10.5465/amr.2013.0350

Hrabia, C.-E., Masuch, N. and Albayrak, S. (2015), "A Metrics Framework for Quantifying Autonomy in Complex Systems", *Multiagent System Technologies / 13th German Conference, MATES 2015, Cottbus, Germany, September 28 - 30, 2015*, Springer, Cham, pp. 22-41. https://doi.org/10.1007/978-3-319-27343-3_2

Huang, H.-M., Pavek, K., Ragon, M., Jones, J., Messina, E. and Albus, J. (2007), "Characterizing Unmanned System Technology: Contextual Autonomous Capability and Level of Autonomy Analyes", *Unmanned System Technology IX, Orlando, Florida*. https://doi.org/10.1117/12.719894

iRobot (2017), *Home Robots: Robot Vacuums*. [online] iRobot. Available at: http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx (accessed 13.11.2017)

Lodwich, A. (2016), "Differences between Industrial Models of Autonomy and Systemic Models of Autonomy", *arXiv:1605.07335 [cs.AI]*.

Murphy, R. R. (2000), *Introduction to AI Robotics*, The MIT Press, Cambridge.

Parasuraman, R., Sheridan, T.B. and Wickens, C.D. (2000), "A Model for Types and Levels of Human Interaction with Automation", *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, Vol. 30, No. 3, pp. 286-297. https://doi.org/10.1109/3468.844354

SAE (2016), *Taxonomy and Definitions for Terms Related to Driving Automation Systems*. [online] SAE International. Available at: https://saemobilus.sae.org/content/j3016_201609 (accessed 13.11.2017).

Schillo, M. and Fischer, K. (2004), "A Taxonomy of Autonomy in Multiagent Organisation", *Agents and Computational Autonomy. AUTONOMY 2003. Lecture Notes in Computer Science, Vol. 2969*, Springer, Berlin, Heidelberg, pp. 68-82. https://doi.org/10.1007/978-3-540-25928-2_6

Sheridan, T.B., Verplank, W.L. and Brooks, T.L. (1978), Human and Computer Control of Undersea Teleoperators, Defense Technical Information Center. https://doi.org/10.21236/ada057655

Thórisson, K.R. and Helgasson, H.P. (2012), "Cognitive Architectures and Autonomy: A Comparative Review", *Journal of Artifical General Intelligence*, Vol. 3 No. 2, pp. 1-30. https://doi.org/10.2478/v10229-011-0015-3

Thrun, S. (2004), "Toward a Framework for Human-Robot Interaction", *Human-Computer Interaction*, Vol. 19, No. 1, pp. 9-24. https://doi.org/10.1207/s15327051hci1901&2_2

Urmson, C. (2015), *The View from the Front Seat of the Google Self-Driving Car*. [online] Backchannel. Available at: https://medium.com/backchannel/the-view-from-the-front-seat-of-the-google-self-driving-car-46fc9f3e6088 (accessed 8.11.2017).

Wakabayashi, D. (2017), *Waymo's Autonomous Cars Cut Out Human Drivers in Road Tests*. [online] The New York Times. Available at: https://www.nytimes.com/2017/11/07/technology/waymo-autonomous-cars.html (accessed 8.11.2017).

Weilkiens, T. (2007), *Systems Engineering with SysML/UML: Modeling, Analysis, Design*, Morgan Kaufmann OMG Press. https://doi.org/10.1016/b978-0-12-374274-2.x0001-6

Wooldridge, M. and Jennings, N. R. (1995), "Intelligent agents: theory and practice", *The Knowledge Engineering Review*, Vol. 10, No. 2, pp. 115-152. https://doi.org/10.1017/S0269888900008122

Yanco, H.A. and Drury, J. (2004), "Classifying Human-Robot Interaction; An Updated Taxonomy", *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, pp. 2841-2846. https://doi.org/10.1109/icsmc.2004.1400763

Zeigler, B. P. (1990), "High Autonomy Systems: Concepts and Models", *Proceedings [1990]. AI, Simulation and Planning in High Autonomy Systems, Tucson, AZ.* https://doi.org/10.1109/aihas.1990.93914

Torben Frans Beernaert, Bachelor of Science
Eindhoven University of Technology, Mechanical Engineering
Ir Kalffstraat 219A, 5617BL Eindhoven, Netherlands
Email: t.f.beernaert@student.tue.nl