Towards a Taxonomy of Autonomous Systems

Stefan Kugele✉[1] [0000-0002-9833-4548], Ana Petrovska[2] [0000-0001-6280-2461], and Ilias Gerostathopoulos[3] [0000-0001-9333-7101]

[1] Technische Hochschule Ingolstadt, Research Institute AImotion Bavaria, Germany
Stefan.Kugele@thi.de

[2] Technical University of Munich, Department of Informatics, Germany
ana.petrovska@tum.de

[3] Vrije University in Amsterdam, Faculty of Science, The Netherlands
i.g.gerostathopoulos@vu.nl

Abstract. In this paper, we present a precise and yet concise characterisation of autonomous systems. To the best of our knowledge, there is no similar work, which through a mathematical definition of terms provides a foundation for describing the systems of the future: autonomous software-intensive systems and their architectures. Such systems include robotic taxi as an example of 2D mobility, or even drone/UAV taxi, as an example in the field of 3D urban air mobility. The presented terms lead to a four-level taxonomy. We describe informally and formally the taxonomy levels and exemplarily compare them to the degrees of automation as previously proposed by the SAE J3016 automotive standard.

Keywords: Autonomous systems · Taxonomy · Architecture

1 Introduction

The world is changing, and so are systems. Woods [9] describes in his much-noticed article "Software Architecture in a Changing World" the evolution from monolithic systems back in the 1980s to intelligent connected systems in the 2020s. We share Woods's vision for future systems. Today's connected cyber-physical systems (CPSs) are not too far away from this vision. The missing link between the current systems and the autonomous systems that we outline for the future is twofold: First, systems will be capable of adapting their structure and behaviour in reaction to changes and uncertainties emerging from their environment and the systems themselves [4,6,8] – they will be adaptive systems. Second, they will be able to derive knowledge themselves during their operational time to infer actions to perform.

The modern CPSs, such as cooperative robotic systems or intelligent transportation systems, are per se distributed. The not too distant future probably brings hitherto unrivalled levels of human-robot interaction. In such scenarios, machines and humans share the same environment, i. e., operational context [1, 4]. Examples for those shared environments are (i) production systems (cf. Industry 4.0) or (ii) intelligent transportation systems with both autonomous and human-operated mobility. As a result, autonomous behaviour becomes an indispensable characteristic of such systems.

The lack of shared understanding of the notion of autonomy makes it difficult for the works across various domains to be compared or even discussed since the same term is used with different semantics. For example, very often in the literature, Unmanned Aerial Vehicles (UAVs) are misleadingly referred to as autonomous, although an end user completely controls their flying operation. As another example, we take robots operating in a room, which use Adaptive Monte Carlo Localisation (AMCL) to localise themselves and navigate in the space. Even though the robots localising and navigating independently in the room is some form of autonomy, they simply cannot be called *fully autonomous systems* if they operate in a room in which they often collide or get in deadlocks. In these situations, human administrators need to intervene in order for the robots to be able to continue with their operation. The intervention from a user (i. e., human administrator) directly affects the system's autonomy.

In response, we present in this paper our first steps towards a unified, comprehensive, and precise description of autonomous systems. Based on the level of user interaction and system's learning capabilities, we distinguish four autonomy levels ($\mathsf{A_0}$-$\mathsf{A_3}$): non-autonomous, intermittent autonomous, eventually autonomous, and fully autonomous. Our goal is to offer a precise and concise terminology that can be used to refer to the different types/levels of autonomous systems and to present a high-level architecture for each level.

The remainder of this paper is structured as follows. In Sect. 2 we briefly sketch existing efforts to formalise autonomy and explain the formal notation we are using later on. In Sect. 3, we present our taxonomy. Finally, in Sect. 4, we discuss and conclude the paper and outline our further research agenda.

2 Background

2.1 Existing Efforts to Formalise Autonomy

An initial effort in the literature to formally define autonomy was made by Luck and d'Inverno [5]. In this paper, the authors argue that the terms agency and autonomy are often used interchangeably without considering their relevance and significance, and in response, they propose a three-tiered principled theory using the Z specification language. In their three-tiered hierarchy, the authors distinguish between objects, agents, and autonomous agents. Concretely, in their definition of autonomy, as a focal point, the authors introduce *motivations*—"higher-level non-derivative components related to goals." Namely, according to their definition, autonomous agents have certain motivations and some potential of evaluating their own behaviour in terms of their environment and the respective motivations. The authors further add that the behaviour of the autonomous agent is strongly determined by and dependent on different internal and environmental factors. Although the authors acknowledge the importance of considering different internal and environmental (i. e., contextual) factors while defining autonomy, in their formalisms, the importance of the user in defining autonomy is entirely omitted. On the contrary, in our paper, we put the strongest emphasis on the user. Concretely, how the involvement of the user in the operation of the system diminishes, proportionally to the increase of the system's autonomy. We define levels of system's autonomy by focusing on the system's function and how much from the user's logic is "shifted" to the system in the higher levels of autonomy. We further touch on the importance of *learning*, especially when 1) the systems operate in highly dynamic, uncertain and unknown environments, and 2) the user's control on the system reduces. To the best of our knowledge, there is no prior work that defines different levels of autonomy *formally*.

2.2 Formal Modelling Approach

Within this paper, we use the formal modelling notation Focus introduced by Broy and Stølen [2]. We restrict ourselves to only those concepts necessary for the understanding of this work. In Focus, systems are described by their (i) *syntactic* and their (ii) *semantic* interface. The semantic interface of a system is denoted by $(I \triangleright O)$ indicating the set of *input* and *output channels*, $I, O \subseteq C$, where C denotes the set of all channels. Systems are (hierarchically) (de-)composed by connecting them via channels. A *timed stream* s of messages $m \in M$, e. g. $s = \langle\langle m_1\rangle\ \langle\rangle\ \langle m_3\ m_4\rangle \dots\rangle$, is assigned to each channel $c \in C$. The set of timed streams $\mathcal{T}(M)$ over messages M associates to each positive point in time $t \in \mathbb{N}^+$ a sequence of messages M^*, formally $\mathcal{T}(M) = \mathbb{N}^+ \to M^*$. In case of finite timed streams, $\mathcal{T}_{\mathsf{fin}}(M)$ is defined as: $\mathcal{T}_{\mathsf{fin}}(M) = \bigcup_{n\in\mathbb{N}} ([1\colon n] \to M^*)$. In the example given, in the first time slot, $\langle m_1\rangle$ is transmitted; in the second time slot, nothing is transmitted (denoted by $\langle\rangle$), and in the third depicted time slot, two messages $\langle m_3\ m_4\rangle$ are transmitted. *Untimed streams* over messages M are captured in the set $\mathcal{U}(M)$ which is defined as $\mathcal{U}(M) = (\mathbb{N}^+ \to M) \cup \bigcup_{n\in\mathbb{N}} ([1\colon n] \to M)$, i. e., each time slot is associated with at most one message and there can be streams of finite length. By $\overrightarrow{C}$, we denote *channel histories* given by families of timed streams: $\overrightarrow{C} = (C \to \mathcal{T}(M))$. Thus, every timed history $x \in \overrightarrow{X}$ denotes an evaluation for the channels in C by streams. With $\#s$, we denote the number of arbitrary messages in stream s, with $m\#s$ that of messages m. For timed streams $s \in \mathcal{T}(M)$, we denote with $s{\downarrow}(t) \in \mathcal{T}_{\mathsf{fin}}(M)$ the finite timed stream until time t. The system's *behavioural function* (semantic interface) f is given by a mapping of *input* to *output histories*: $f\colon \overrightarrow{I} \to \wp(\overrightarrow{O})$.

3 A Taxonomy for Defining Autonomy

In this section, we first describe how autonomy of a system is related to autonomy of its functions, then present the main ideas behind our proposed taxonomy, and finally describe both informally and formally the different levels of autonomy.

3.1 Autonomy as a Property of Individual Functions

CPSs such as modern cars are engineered in a way to deliver thousands of customer or user functions. These are functions that are directly controlled by the user, or at least the user can perceive their effect. Switching on the radio, for example, results in music being played. This is a customer function. On the other hand, there are functions, for example, for diagnosis or for offering encryption services, which the customer cannot control directly, of whose existence often nothing at all is known and whose effects are not visible to the user. Considering the above-mentioned, it is not trivial to classify a complete system as autonomous or non-autonomous. Instead, autonomy is a property of individual functions. Let us take a vehicle that drives autonomously. We assume that this system still offers the functionality to the passengers to choose the radio station or the playlist themselves. Thus, the CPS operates autonomously in terms of driving but is still heteronomous in terms of music playback. A similar argumentation applies, for example, to vehicles that are equipped with automation functions of varying degrees of automation, as considered in the SAE J3016 standard. For this system, as well as for other multi-functional systems, it is not meaningful to conclude from the autonomy of a single function, the autonomy or heteronomy of the whole system. Therefore, the commonly used term of an autonomous vehicle is too imprecise since the term autonomy refers exclusively to its driving capabilities. Hence, also the SAE proposes not to speak about "autonomous vehicles" but instead about "level [3, 4, or 5] Automated Driving System-equipped vehicles" (cf. [7], §7.2).

The only two statements that can be made with certainty are the following: (1) if all functions of a system are autonomous, then the system can also be called autonomous, and (2) if no function is autonomous, then certainly the system is not autonomous. Anything in between cannot be captured with precision. Single-functional systems are a special case. In such systems, the autonomy or heteronomy of the single function is propagated to the system. For the sake of illustrating our taxonomy on a simpler case, we will focus on single-functional systems in the rest of the paper.

3.2 Main Ideas Behind the Taxonomy for Autonomy

Our first main idea is to define autonomy levels of a system by focusing on the system's function and specifically by looking at the level of interaction that a user has with the system. Intuitively, the more user interaction is in place, the less autonomous the system is. "More user interaction" can mean both more *frequent* interaction and more *fine-grained* interaction. Actually, these two characteristics very often go hand in hand: consider, for instance, the case of a drone: it can be controlled with a joystick with frequent and fine-grained user interaction (lower autonomy); it can also be controlled via a high-level target-setting routine with less frequent and more coarse-grained user interaction (higher autonomy).

The second main idea behind our taxonomy is to distinguish between systems that learn and ones that do not learn. By learning, we mean that systems can observe both their context and user actions and identify behavioural patterns (e. g. rules or policies) in the observed data (e.g. by training and using a classifier). Such patterns can be used at run-time to reduce the amount of user interaction with the system gradually. Hence, the more capable a system is of learning behavioural patterns, the more autonomous it can become.

Finally, the third main idea is to define a system as autonomous within an assumed operational context. The assumed context can be narrow (e. g. a drone operating in a wind range of 0-4 Beaufort) or very broad (e. g. a drone operating under any weather conditions). The specification of the context can also be uncertain or incomplete, i. e., the designers of the system might not be able to anticipate and list all possible situations that may arise under a specific context assumption. In any case, the more broad context is assumed, the harder it becomes for a system to reach high autonomy.

3.3 Taxonomy Levels

The four levels of autonomous systems in our taxonomy are shown in Fig. 1. Figure 2 shows the interaction between the user u, the context c, and the system s, as well as the (very high level) architecture of the system at each level in the taxonomy.

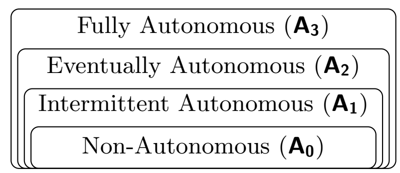
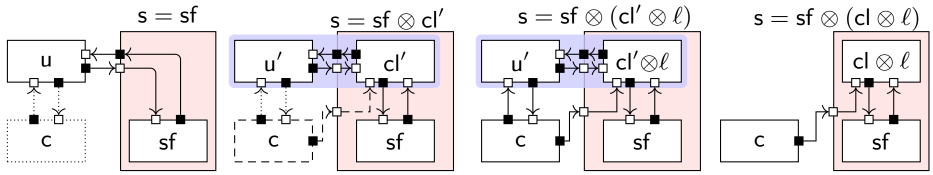


Fig. 1: Taxonomy levels.

The lowest level, $\mathbf{A_0}$, refers to systems that are *not autonomous*. For these systems, user input is needed at all times for controlling their operation. Examples are using the radio in a car or controlling the movement of a robot via a remote controller. As can be seen in Fig. 2(a), on this level, the system s (i. e., the system function sf) is completely controlled by the user and does not assume any input from the context (although this input might be already taken indirectly into account by the user). Note that the function sf might internally do something in the background that does not depend on the user input. A user can control the movement and trajectory of a drone; however, each drone internally provides attitude stabilisation that is not dependent on user input but is part of this system function.

The next level, $\mathbf{A_1}$, refers to systems that are *intermittent autonomous*: they can operate autonomously in-between two consecutive user inputs. In this case, the system can receive user input periodically or sporadically. As shown in Fig. 2(b), part of the logic of the user is shifted to the system as a control logic cl', which interacts with the system function sf. Input to the control logic can also be provided by the context. For instance, consider the movement of a robotic vacuum cleaner: the system perceives its environment through its sensors (obtains context input) and operates autonomously until it gets stuck (e. g. because of an obstacle or a rough surface); at this point, a user is required to intervene to restart the robot or point it to the right direction.

Level $\mathbf{A_2}$, shown in Fig. 2(c), refers to *eventually autonomous* systems: here, the user interaction reduces over time until the system reaches a point where it does not require any user interaction (user control). For this to happen, the system's control logic cl' is usually enhanced and equipped with a learning component ℓ that is able to identify the user interaction patterns associated with certain system and context states. An example is a robotic vacuum cleaner that is able to learn how to move under different floor types (e. g. faster or slower) and avoid crashes that would necessitate user interaction. Clearly, the degree and sophistication of monitoring and reasoning on context changes and user actions is much higher than in intermittent autonomous systems.



(a) A_0 (b) A_1 (c) A_2 (d) A_3

Fig. 2: From user-operation to autonomy: (a) A human user u controls the system s (i. e., the system's function sf). (b) The control logic is divided between the user u' and the system cl', i. e., $\mathsf{u} = \mathsf{u}' \otimes \mathsf{cl}'$. (c) The control logic of the system cl' could be enhanced with a learning component ℓ to better address e. g. changes in the context c. (d) The control logic cl with the usually necessary learning component ℓ is entirely performed by the system itself.

Finally, level A_3 refers to *fully autonomous* systems, where no user input is needed (except the provision of initial strategic or goal-setting information), as it can be seen in Fig. 2(d). Systems on this level of autonomy can observe and adjust their behaviour to *any* context by potentially integrating learning in their control logic cl. Please note that the necessity and the sophistication of the learning is proportionate to 1) the complexity and the broadness of the context, and 2) the specifications of the context in the systems, as previously explained in Sect. 3.2. For instance, a robotic vacuum cleaner can move in a fully autonomous way when its context is more simplistic and could be fully anticipated (e. g. prescribed environment that contains only certain floor and obstacle types). To achieve this, the system needs to be equipped with sensing and run-time reasoning capabilities to adjust its movement behaviour and remain operational without human interaction. However, the difficulty for the same system to remain fully autonomous increases proportionally to the complexity of its context. For example, the context can be dynamic in ways that could not be anticipated, resulting in uncertain and incomplete context specifications. Since the user on this level is entirely out of the loop, this would require new, innovative, and more sophisticated learning methods in the *fully autonomous* systems.

We note that one can also imagine relatively simple systems without context impact that are configured *once or not at all* by a user and then work *without any user interaction or learning* (e. g. an alarm clock); while these systems also technically fall under A_2 or A_3, they are less complex and sophisticated.

3.4 Formalisation of Taxonomy Levels

The intuitively described taxonomy levels are specified mathematically in the following. We denote with u the input stream from the user to the system.

Definition 1 (Non-Autonomous, A_0). *A system is called* non-autonomous*, iff it solely depends on user inputs:* $\forall t \in \mathbb{N}^+ : \mathsf{u}(t) \neq \langle\rangle$.

If there is less and less intervention or input by users, this becomes necessary repeatedly; we speak of intermittent autonomy.

Definition 2 (Intermittent Autonomous, $\mathsf{A_1}$). *A system is called* intermittent autonomous*, iff user interaction is necessary from time to time (periodic or sporadic), i. e.:* $\forall t \in \mathbb{N}^+\ \exists t', t'' > t, t', t'' \in \mathbb{N}^+, t' \neq t'' \colon \mathsf{u}(t') \neq \langle\rangle \wedge \mathsf{u}(t'') = \langle\rangle$.

We emphasised that *learning* is essential in order to reach even higher levels of autonomy. By learning, the system converges to a point t after which no user interaction is needed anymore. Such systems are called *eventually autonomous*.

Definition 3 (Eventually Autonomous, $\mathsf{A_2}$). *A system is called* eventually autonomous*, iff after time* $t \in \mathbb{N}^+$ *no user input or intervention is needed anymore to fulfil the mission goals:* $\exists t \in \mathbb{N}^+ \colon \forall t' > t \colon \mathsf{u}(t') = \langle\rangle$.

In other words, only a finite number n of messages were transmitted up to t and no further messages will be transmitted beyond that time: $\#\mathsf{u}{\downarrow}(t) = n$, with $n \in \mathbb{N}$. The smaller t is, the earlier the point of autonomy is reached. If this is already the case from the beginning, we speak of *fully autonomous* systems.

Definition 4 (Fully Autonomous, $\mathsf{A_3}$). *A system is called* fully autonomous *if no user interaction or intervention is necessary at all, i. e.,* $\forall t \in \mathbb{N}^+ \colon \mathsf{u}(t) = \langle\rangle$.

Eventual and full autonomy make strict demands on the ability to precisely perceive and analyse the context, and draw conclusions and learn from it. However, in many respects, it will probably not be possible to achieve them in the foreseeable future for a not highly restricted operational context. Reasons for this are manifold and include the limited ability to fully perceive and understand the context and be prepared for all conceivable operational situations. Therefore, let us now consider intermittent autonomy. Assume the case that every other time step (e. g. every second minute), there is user interaction on an infinite timed stream, see u_1 below. This results in an infinite number of interactions. In another case, there could be one interaction every millionth minute, as shown in u_2. These two cases are equivalent or indistinguishable by definition.

$$\mathsf{u}_1 = \langle\langle m\rangle\langle\rangle\langle m\rangle\langle\rangle \dots \langle m\rangle\langle\rangle \dots\rangle,\ \mathsf{u}_2 = \langle\langle m\rangle\langle\rangle^{10^6-1}\langle m\rangle\langle\rangle^{10^6-1} \dots \langle m\rangle\langle\rangle^{10^6-1} \dots\rangle$$

This is due to Cantor's concept of infinity. Intuitively, however, a system that depends on user input every two minutes acts less autonomously than a system that can operate for almost two years (1.9 years in u_2) independently. Therefore, intermittent autonomy extends from "almost" no autonomy towards "almost" eventually autonomy. The classification in this spectrum can be made more precise if we take a closer look at the frequency of user input. Because of the above discussion on infinity, we only consider prefixes of finite length of (in)finite streams, i. e., $\mathsf{u}{\downarrow}(t)$. Let $\alpha \in (0, 1)$ be the ratio between times without user input and the interval $[1; t]$, i. e., $\alpha = \langle\rangle\#\mathsf{u}/t$. The closer α gets to one, the more autonomous the system is.

4 Discussion and Conclusion

Comparison to SAE Levels (L0-L5) [7]. No driving automation (L0) refers to $\mathsf{A_0}$–*no autonomy*, L1/2 (driver assistance, partial driving automation) can be defined with the notion of intermittent autonomy–$\mathsf{A_1}$, conditional driving automation (L3), applies for $\alpha \approx 1$ in a limited operational context such as highway autopilots. Finally, high driving automation (L4) and full driving automation (L5) are captured by our level $\mathsf{A_3}$, *full autonomy.* For both, different assumptions, w.r.t. the context or the operational design domain, need to be made.

Future Extensions. It would be relevant to investigate the relation between the higher levels of autonomy and *self-** properties (cf. [3]) of the systems, e. g. self-adaptation. In our current understanding, adaptivity is a precondition for a higher autonomy since it enables the system to deal with various unanticipated changes and uncertainties; however, a clear distinction and definition of these two notions is still open. Another open issue refers to the notion of messages exchanged in intermittent autonomous systems. We have tried to distinguish between two intermittent autonomous systems based on their frequency of message exchange, but the expressiveness of messages is also important. Not every message has to have the same "information content". It is a matter for future research and discussion whether this point can be captured using, e. g. Shannon's definition of information content (a limitation of this approach is the assumption of statistical independence and idempotence of messages). To what extent or when is this a permissible limitation is an open question.

Conclusion. In this paper, we proposed a taxonomy that supports the formal specification of different levels of autonomous systems. We have also proposed a high-level architecture for each level to exemplify the user, context, and system interaction. Our goal is to propose a terminology that, if broadly accepted, can be used for more effective communication and comparison of autonomy levels in software-intensive systems that goes beyond the well-known SAE J3016 for automated driving.

References

1. Broy, M., Leuxner, C., Sitou, W., Spanfelner, B., Winter, S.: Formalizing the notion of adaptive system behavior. In: ACM Symposium on Applied Computing (SAC). pp. 1029–1033. ACM (2009)
2. Broy, M., Stølen, K.: Specification and Development of Interactive Systems–Focus on Streams, Interfaces, and Refinement. Monogr. in Comp. Science, Springer (2001)
3. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
4. de Lemos, R., et al.: Software engineering for self-adaptive systems: A second research roadmap. In: Software Engineering for Self-Adaptive Systems. LNCS, vol. 7475, pp. 1–32. Springer (2010)
5. Luck, M., d'Inverno, M.: A formal framework for agency and autonomy. In: First International Conference on Multiagent Systems. pp. 254–260. The MIT Press (1995)
6. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM transactions on autonomous and adaptive systems (TAAS) **4**(2), 1–42 (2009)
7. Society of Automotive Engineers: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, SAE j3016 (2018)
8. Weyns, D.: Software engineering of self-adaptive systems. In: Handbook of Software Engineering, pp. 399–443. Springer (2019)
9. Woods, E.: Software architecture in a changing world. IEEE Softw. **33**(6), 94–97 (2016)