Article

Fundamental Research Challenges for Distributed Computing Continuum Systems

Victor Casamayor Pujol * , Andrea Morichetta , Ilir Murturi and Praveen Kumar Donta
and Schahram Dustdar

Distributed Systems Group, TU Wien, 1040 Vienna, Austria; a.morichetta@dsg.tuwien.ac.at (A.M.);
i.murturi@dsg.tuwien.ac.at (I.M.); p.donta@dsg.tuwien.ac.at (P.K.D.); dustdar@dsg.tuwien.ac.at (S.D.)
* Correspondence: v.casamayor@dsg.tuwien.ac.at

Abstract: This article discusses four fundamental topics for future Distributed Computing Continuum Systems: their representation, model, lifelong learning, and business model. Further, it presents techniques and concepts that can be useful to define these four topics specifically for Distributed Computing Continuum Systems. Finally, this article presents a broad view of the synergies among the presented technique that can enable the development of future Distributed Computing Continuum Systems.

Keywords: distributed computing continuum systems; system representation; system model; lifelong learning; business model

1. Introduction

The expansion from Cloud computing to Edge computing has brought a new paradigm called the Distributed Computing Continuum [1–4]. This combines the virtually unlimited resources of the Cloud with the heterogeneity and proximity of the Edge. To do so, the Distributed Computing Continuum combines the underlying infrastructure of all other computing tiers. Hence, the infrastructure becomes a first-class citizen compared to current Internet-distributed systems.

Current research on Edge computing and Distributed Computing Continuum focuses on solving specific problems, which produce particular solutions with narrow applicability. A few examples include: in [5], the approach is tailored to an ultradense network; in [6], the authors present a solution for a static description of the system; or in [7], the authors present an orchestration for the edge-cloud that requires centralization on the Cloud. In this article, we aim to show pointers to generalized solutions; we organize it through the highlighting of four key aspects that require in-depth analysis, as well as a high degree of agreement among the scientific community and the other stakeholders to make tangible progress on the development of the Distributed Computing Continuum.

First, the Distributed Computing Continuum needs a novel **representation** beyond the classical architecture of computer systems. Distributed Computing Continuum Systems are built of a large variety of heterogeneous devices and networks. The system's functional requirements can either naturally evolve during their lifetime, dynamically change the running services [8], or, more critically, suffer unexpected events. These changes will affect the underlying infrastructure configuration, out-dating previous architectural representations. For instance, the Edge infrastructure requires a dynamic adaption to new devices and network connections, leading to a completely new system from the perspective of its design phase. This behavior contrasts with Cloud computing, where changes in the underlying infrastructure can be updated, but the application is not affected.

Another challenge for Distributed Computing Continuum Systems is their **model**. First, we need to clarify the difference between the representation and the model. Our view

is that representation is a description of the system, its components, its relations, and its characteristics. Whereas we address the dynamic behavior of the system and its components by the model. Nevertheless, both concepts require a certain level of agreement as, ideally, one aims to have a compatible representation and model of the system. The complexity of the system, coupled with its openness (i.e., many external and spontaneous actors can affect the system), endangers the correctness of adaptation strategies; in the Cloud, this is usually solved by only considering a single elasticity strategy per component [9]. Further, it is complex to assess the impact of the adaptation on the entire system, i.e., using a different set of Edge devices might imply moving data through another network, which can affect privacy/security constraints. Hence, tools to describe the new behavior of the infrastructure are needed.

The third key element is a **lifelong learning framework**. The dynamicity of the environment, the user's variety of behaviors, the evolution of functional requirements, and the long-term usage of the underlying infrastructure require developing a learning framework to keep high-quality standards during the system's entire life cycle. This is aligned with the idea presented in [10] about lifelong learning for self-adaptive systems.

The last facet that requires agreement among the community is the **business model**. A key enabler for the Cloud tier has been its successful business model. Nevertheless, the multi-tenant and multi-proprietary characteristics of the underlying infrastructure produce a more challenging set of stakeholders for Distributed Computing Continuum Systems. However, to attract the needed collaborations and investments to develop such an ambitious computing tier, it is of utmost importance to develop certain agreements that can enable the best business model. In this regard, we have witnessed, in the context of the Mobile World Congress 2023, how the big telecommunication companies want to be part of the Distributed Computing Continuum Systems by providing an API to application developers to tailor their networks to the application's needs, e.g., see "GSMA Open Gateway", from Future Networks (https://www.gsma.com/futurenetworks/?page_id=35 168—accessed 1 March 2023).

The main goal of this work presented as a vision for the emerging Distributed Computing Continuum Systems, is to highlight the need for a holistic perspective. This type of system is far from being a reality, and we believe that common grounds are required to advance their development. Hence, we provide what we think are fundamental research challenges to be solved and what are our research road-maps for each of them. We seek to spark discussion and creativity in the research community to enable these future systems.

To sum up, Distributed Computing Continuum Systems require a broad agreement on a representation, a model, a lifelong learning framework, and a business model to enable its development. This article presents a few ideas on how to start building these required blocks. In the following section, a technique or concept is presented for each of the presented aspects to shape our vision of Distributed Computing Continuum Systems. Then, we discuss the overall merging of all presented concepts and techniques, and we finish this article with a conclusion and future work.

2. Vision

In this section, we present key elements for each of the previously introduced aspects that, from our perspective, will be key for developing Distributed Computing Continuum Systems.

2.1. Representation

The characteristics of the Distributed Computing Continuum require new representations for Internet-distributed systems enabling dynamic systems and topologies. This contrasts with the usage of the concept of system architecture, where the word architecture refers to a static structure of the system. As previously discussed, the complex and dynamic behavior of Distributed Computing Continuum calls for other techniques to represent these systems, which can accommodate the dynamic behavior of the underlying infrastructure and the system's environment.

Further, a fundamental concept requires an in-depth discussion: the definition and scope of the system and its relation to each component. Simply put, the Distributed Computing Continuum needs to be understood as an ecosystem in which there are different abstraction layers, where components are described and aggregated differently. Interestingly, when one thinks about an ecosystem, the synergies and dependencies between the components blur the definition of self, i.e., the boundaries of an autonomic component are flexible and dependent on the purpose. Components interact with others both horizontally and vertically with respect to their abstraction levels. In this regard, Distributed Computing Continuum Systems have multi-level and multi-scale structures, and their components show a dual tendency; from one side, they aim to be autonomous, and from the other side, they need to be integrated with others to provide a complete view of the system. Hence, they require defining what is the *self*: the entire system, a single autonomic component, or all things simultaneously. Hence, we need to analyze and provide arguments for each case to understand which is the best solution for the system's representation.

Our initial intuition is that we need a holistic view of the system, considering it as a system of systems and providing compatible tools at any level. Hence, we envision the **Markov Blanket** as a key element to represent Distributed Computing Continuum Systems given its nesting and filtering capacity [11].

Markov Blanket

The Markov Blanket, in Bayesian statistics, refers to the set of variables that contain all needed information to determine a target variable. Simply put, the Markov Blanket concept can be used to determine which variables influence another. Formally, if x is a random variable, and Y is the set of random variables of the Markov Blanket of x, then $P(x|Y) = P(x|Y, Z)$, where Z represents any other random variable [12].

In large-scale Distributed Computing Continuum Systems, the Markov Blanket concept can be seen as causal filtering, given that it allows working only with the subset of variables affecting the target. This is key in terms of scalability, e.g., the problem of selecting the best device for each service in an application is NP-Hard with exponential time complexity [13]; hence dealing with only the required subset of components can drastically alleviate the difficulty. Further, the definition of a random variable within the Markov Blanket scope is flexible, which means that regardless of the abstraction level in which the variable exists, it will be possible to build its Markov Blanket. Hence, we can foresee this as a nesting capability in which the higher-level abstraction Markov Blanket can be decomposed as a set of other Markov Blankets at lower-level abstractions. If we bring this to the Distributed Computing Continuum Systems, it is possible to define the entire system with a Markov Blanket describing the main components that affect the system, and it is possible to go deeper in detail and describe smaller components also in terms of a Markov Blanket. Hence, both scalability and the self's definition are addressed by using the Markov Blanket abstraction. In conclusion, the Markov Blanket concept is needed to represent Distributed Computing Continuum Systems.

2.2. Model

Two main challenges are identified specific to Distributed Computing Continuum Systems to develop models of their behavior. (1) Its decentralization precludes developing a model of the entire system as a single entity; conversely, the model should allow its distribution. In other words, the model has to allow its splitting among the different parts of the system. Further, it can be linked with the representation; one could imagine that there is a model for each Markov Blanket used to represent the system. (2) Distributed Computing Continuum Systems are set within an uncertain environment. On the one side, there is unpredictability in the user's behavior, e.g., in an autonomous vehicle use case, external computations require to follow a car through its trajectory to keep latency at a lower bound [14], but the required hops can not be predicted beforehand, given that the trajectory can change on demand. On the other side, the underlying infrastructure of

Distributed Computing Continuum Systems is multi-tenant and multi-proprietary. Hence, another source of uncertainty is from the usage of others tenants of the shared resources. Further, their complexity and interconnection also generate internal uncertainty, given that the exact knowledge of the system's behavior might not be known; in this regard, there are many sources of internal uncertainties for self-adaptive software systems, as explained in [15], in brief, the model has to handle uncertainty.

To develop the system's model, we look at one concept and one technique. The concept, called **DeepSLOs**, can define and link constraints for a Distributed Computing Continuum System at different abstraction levels. In contrast, the technique, **Causal inference** allows the system to understand its own behavior and to perform a priori analyzed changes on its underlying infrastructure to minimize the effects of uncertainty.

2.2.1. DeepSLOs

DeepSLOs stem from Cloud Service Level Objectives (SLOs). In general, an SLO is a constraint to the underlying infrastructure, simply put, the minimal availability of a service or the maximal CPU usage of a workload. Hence, in the Cloud, when a constraint is violated, an elasticity strategy is triggered, which modifies the system and brings the Service Level Indicator (SLI) within the value specified by the SLO. In contrast, a DeepSLO is a set of hierarchically connected SLOs. This deviates from Cloud SLOs because they are no longer isolated constraints of the system. However, they provide a holistic perspective on the system status given by its constraints. Hence, it is possible to obtain a complete description of the system's performance. Further, DeepSLOs also aim to fully describe the characteristics of the underlying infrastructure, which highlights the infrastructure as a key part of any Distributed Computing Continuum System. Within a single DeepSLO, there are SLOs at different levels of abstraction, which describe the system's performance from different perspectives. Simply put, low-level abstraction SLOs can easily define infrastructure behavior, i.e., the performance of GPUs at the Edge, while higher-level abstraction can deal with application performance, i.e., the accuracy of the inference job at the Edge. Both given examples can be expressed as constraints within a single DeepSLO, each as an SLO but connected through the system dependencies. Still, the first brings information on the hardware performance for the inference job, and the second has a holistic performance on the tasks. In this way, different elastic strategies can be used efficiently depending on the cause of the system's performance degradation.

2.2.2. Causal Inference

Causal inference [16] is a mathematical framework able to discover causal relations between system variables and predict their behavior. Hence, it provides a better understanding of why events occur, solving conflicts when several adaptation strategies are needed. Further, it can predict the outcome of interventions in the system. Simply put, it predicts how the system will behave after a new configuration. Finally, causal inference unfolds the ability to develop counterfactuals to better understand the system's true behavior, providing a key element to the learning capability. On the one side, studying interventions, i.e., applying a specific condition to the variables, can ease the selection of the adaptation strategies; further, using counterfactuals, hypothetical situations can be described to extract learnings and improve the system's resistance against uncertainty. These methods are very relevant for uncertainty management, given that it is possible to achieve knowledge of the system's behavior under conditions not yet met in real operations. Causal inference is usually applied over a directed acyclic graph to perform its analyses and explain the behavior of the system; interestingly, this graph can be provided by the Markov Blanket representation of the system.

One can imagine an application with a peak of demand in some region. Hence, the Distributed Computing Continuum System is required to perform an adaptive action to properly maintain its expected quality standards. However, there can be several ways to tackle that situation, scaling up Cloud or Edge components, using new resources, etc. In

such situations, causal inference provides an understanding of how the system will react to these changes, e.g., how costs can be increased, how other services can be affected, how sustainable is the solution, etc. Further, if quality and cost (see Section 2.4) are not fixed, and there are some assumable margins, the manifold of options and their consequences grows. This complexity, also explained through an illustrative use case in [17], can be tackled by a better understanding of the internal system relations provided by causal inference.

2.3. Lifelong Learning Framework

Under the assumption that Distributed Computing Continuum Systems are complex and inhabit a dynamic environment, providing the system with the learning capacity is of foremost importance. Hence, lifelong learning enables improving the models that govern the system, and, due to the nature of these systems, it also needs to be related to composition, i.e., which are the best components of the underlying infrastructure to use in an application. Systems need to be able to change their configuration using other components that are not initially part of the system; in other words, the use of adaptation strategies requires learning, given the huge space of possible configurations. Consequently, they need to make decisions dependent on the system's current state, which cannot be foreseen at design time. Hence, Distributed Computing Continuum Systems need techniques that provide this capacity for continuous improvement in a dynamic setting. In this regard, the Free Energy Principle (FEP) explains the behavior of systems to adapt to their environment, initially developed as a hypothesis on how the brain works [18]. To do so, the FEP aims to maximize the knowledge of the system about its environment, which is needed to continuously improve operation.

Free Energy Principle and Active Inference

The FEP was first defined by K. Friston to describe the behavior of the brain as a system that adapts to its dynamic environment. In brief, the FEP shows that adaptive systems have an internal model of their environment and that this allows adaptive systems to persist in it. Further, the FEP claims that this adaptivity is achieved by minimizing the difference between the internal model that the system possesses of its environment concerning the real environment behavior. Another important observation of the FEP for Distributed Computing Continuum Systems is that regardless of the scale, adaptive systems behave similarly [19].

The active inference is a corollary of the FEP, providing a methodology to develop agents (systems) that can learn from actions following the FEP [20]. Hence, by adapting Distributed Computing Continuum Systems to the active inference methodology, it will be possible to develop systems that progressively learn to adapt within a dynamic environment.

2.4. Business Model

We have learned from Cloud computing that a key aspect of the success of Distributed Computing Continuum Systems is considering and easing their business logic. These systems are multi-tenant and multi-proprietary; hence, we need concepts and methods that allow several stakeholders to collaborate or share part of their infrastructure. We are not considering here which has to be the business model, but two ideas are needed for its emergence. Our intuition is that to enable this, communication and understanding among them need to be the cornerstone. Further, the fact that systems, components, and other involved stakeholders understand different abstractions is a great challenge that needs to be addressed. From this perspective, the use of **Resources**, **Quality**, and **Cost** as the highest-level abstraction state variables enables homogenization among components and systems [21]. Further, they are understandable to all stakeholders given their higher level of abstraction [22].

Finally, security is key to engaging stakeholders. Therefore, **Zero Trust** [23] concepts promise control network flows between all assets, advanced resource protection, fast detection of malicious activities, improved system performance, and secure communication

between components. Features that are vital to make Distributed Computing Continuum a reality.

2.4.1. Resources, Quality, and Cost

In previous work on Cloud computing, Resources, Quality, and Cost have been defined as elasticity dimensions [24], which can define the system's overall state. For Distributed Computing Continuum Systems, we are convinced that the same abstractions are needed; however, they are not elasticity dimensions but the highest-level abstractions of the system state, which will ease the system's management decisions. In other words, Resources, Quality, and Cost are required to understand the system's current situation at the highest level of abstraction and how it can deviate toward other possible states. It is important to remark that Resources, Quality, and Cost are chosen because any system stakeholder can interpret them, and agreements at the highest level can be reached, which then can be specified into lower-level agreements tailored to the specific characteristic of each stakeholder.

2.4.2. Security through Zero Trust

Trustworthiness is key among stakeholders, and secure systems are needed to enforce trust. In this regard, Zero Trust [23,25] is an emerging security paradigm where trust among system components needs to be achieved at every step. In other words, it is not enough to be part of the network's system, but the components' behavior is also continuously verified.

3. Discussion

This section first presents some alternative directions to the ones discussed and shows our holistic vision of the ideas and techniques presented. The alternative ideas presented are not exhaustive as this work presents a vision. Hence, this section aims just at providing a broader context to the current research around Distributed Computing Continuum Systems.

3.1. Techniques Discussion

In terms of representation techniques for Distributed Computing Continuum Systems, some works focus on graph-based representations; in [26], the authors specify the task type and system with graphs and leverage such a representation to solve the task offloading problem. Another perspective is given by [27], where their graph-based representation of Distributed Computing Continuum Systems is the input for graph-based neural networks to solve the distributed scheduling problem. Conversely, our representation aims at being general for whichever problem needs to be solved in the Distributed Computing Continuum paradigm. Further, the filtering and nesting capacity of the Markov Blanket, which is also graph-based, is of utmost importance for Distributed Computing Continuum Systems. Take, for instance, the work in [13], which uses propositional logic to represent the Distributed Computing Continuum System; they show how the complexity of solving the task assignment problem is exponential. Hence, a representation able to filter out the useless aspects for the specific case can make it feasible.

There are several approaches that propose a model for the entire Distributed Computing Continuum that can be leveraged to describe and forecast its expected behavior. Most of the time, the description is left to deep neural networks [28–30], which, in general, provide good results, but it is hard to assess its generalization capabilities. There are also approaches that provide a function-based description that allows for solving the problem, e.g., through game theory [31]. This type of model usually requires many parameters and assumptions on these parameters. A similar situation is found when modeling through queue theory [32,33]. From our perspective, queue theory embraces the randomness found in these systems. However, it requires precise modeling for the probability distributions governing the system behavior. Another interesting approach follows the osmotic computing concept [34,35], which uses the analogy of physical pressure to manage Distributed Computing Continuum Systems. Finally, there are also constraint-based approaches for

system modeling, such as in [36], where they are able to verify the system's performance. We are proposing a constraint-based perspective from the use of DeepSLOs where we can relate the underlying infrastructure with the application to define the entire Distributed Computing Continuum System. Still, we understand that the complex and stochastic behavior requires another modeling layer. There, we see fit for causal inference to leverage a probabilistic model with causal relationships.

We relate lifelong learning with the capacity of a system to continue learning and improving its behavior during its life-cyle when the requirements and tasks can change over time. This problem is usually tackled through machine learning, specifically with reinforcement learning in robotics [37,38]. In this regard, an interesting framework for self-adaptive systems is provided by [10]. A new and interesting approach is on graph continuous learning; a survey is presented in [39]. Our intuition to propose active inference from the FEP relates to its ultimate goal of modeling the dynamic environment to optimize the system's adaptive capacity. Further, its mathematical formulation fits well with the use of Markov Blankets, allowing us to easily combine both techniques. We are aware that FEP has been questioned in terms of applicability and generality [40,41]. Nevertheless, its fitness to our scope and the results showed in other articles [42] are promising.

Regarding the business model, there is a vast amount of work on security aspects, where Zero Trust is one of the most prominent perspectives. In terms of high-level abstractions to set common ground on business objectives, the literature is scarce. We propose Cost, Quality, and Resources that stem from previous work and have worked as expected in Cloud systems. In general, objectives can be QoS parameters, but there is no consensus on how to abstract or normalize such metrics.

3.2. Holistic Perspective

The four key aspects discussed in this article are presented in an integrated view in Figure 1.

The Markov Blanket concept provides a nested representation of the system, which allows us to represent the highest-level state of the system (Resources, Quality, and Cost) as a wrapping blanket over the lower-level states. Interestingly, maintaining the Markov Blanket **representation** across abstraction layers enables the usage of the same techniques, regardless of the abstraction layer. Further, SLOs can leverage this structure of blankets to add constraints on the infrastructure at different layers, covering the entire Distributed Computing Continuum System. Hence, they can be understood as the hooks to the infrastructure, allowing the building of a modular and adaptive system's **model**. Combining these SLOs enables the building of the DeepSLO construct. Hence, by means of causal inference, we can define and predict how the relations within a DeepSLO will behave. It will be possible to tailor the adaptive measures to the predictions through the study of interventional situations. Further, the counterfactual capacity of causal inference can generate hypothetical situations that can be input for a lifelong learning framework. Within this framework, **lifelong learning** can be achieved by using methodologies such as active inference from the FEP that will be biased toward actions that can improve the system's knowledge of its environment. Finally, understanding and trustworthiness are required to build a **business model** for Distributed Computing Continuum Systems; hence, both state variables, such as Resources, Quality, and Cost, as well as a secure environment, are key to that development.
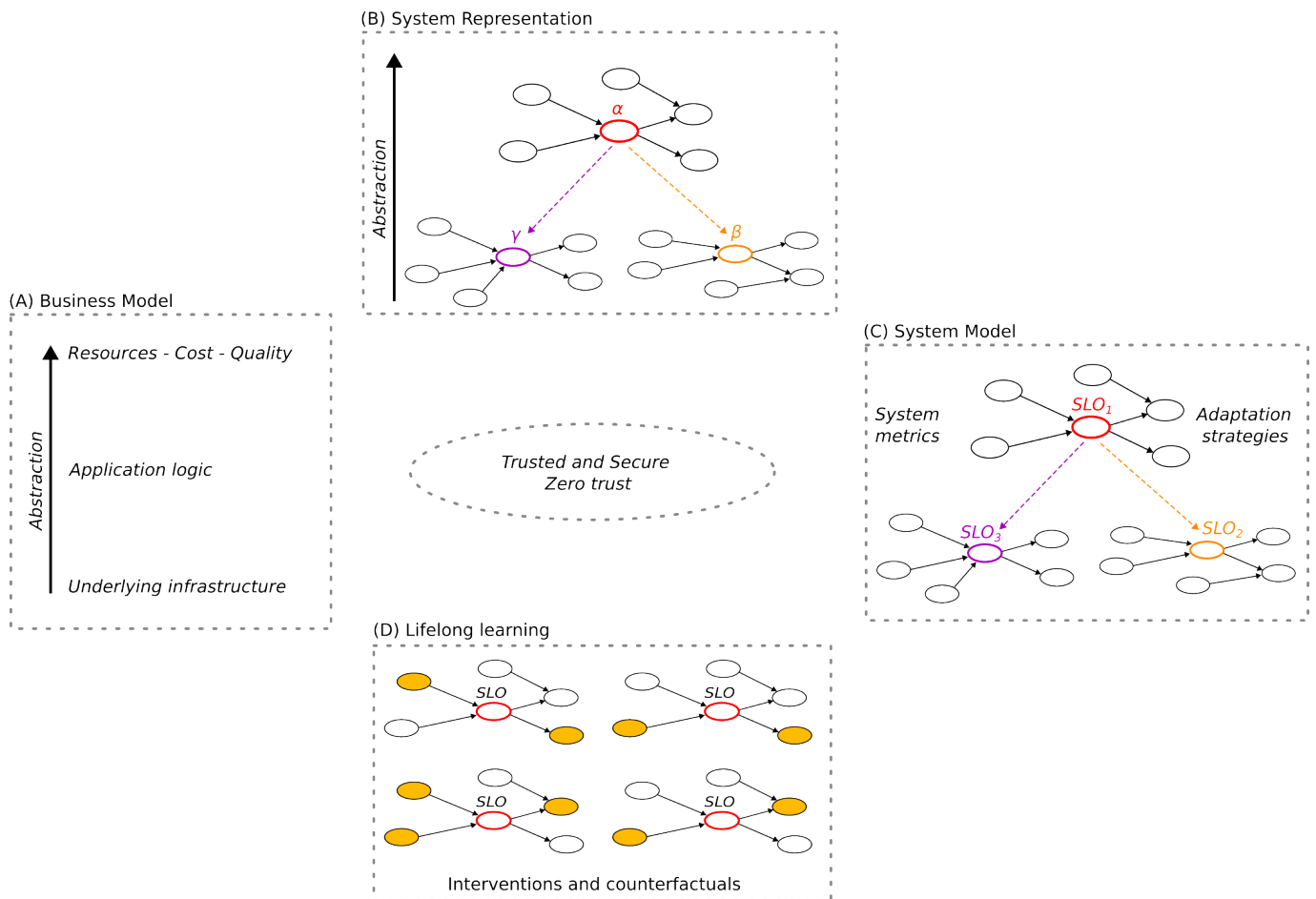
Figure 1. Starting at the center of the figure, we see that all developments require a trusted and secure environment, where Zero Trust techniques will have great relevance due to the heterogeneity and distribution characteristics of the system. On the left (**A**), three different abstractions for a Distributed Computing Continuum System are depicted (the granularity shown can be refined by showing network characteristics or unfolding aspects from the application logic). Each abstraction can easily engage a kind of stakeholder. However, the highest level: Resources-Cost-Quality, aims at being understood by all of them. On top (**B**), there is the representation of the system through Markov Blankets, considering α as a high-level abstraction variable that we can assess by observing the surrounding node, i.e., those factors that affect its value, remarkably thanks to the Markov Blanket approach only those that are relevant are there. Further, α can be decomposed on other lower-level variables (β and γ), providing a nested structure that can cover the entire system. On the right (**C**), we take advantage of this representation to embed SLOs, more specifically, a DeepSLO, in order to model the behavior of the system. As an example, SLO_1 can be related to the quality of a machine-learning inference component, which has its own metrics and adaptation strategies. This is linked to two other lower-level SLOs; in this example, SLO_2 can control the input data by making sure that it has the expected resolution, while SLO_3 controls the required time to perform the inference. Having this decomposition enables a fine-grained capacity for adaptation; other lower-level or higher-level SLOs could be developed if needed to have a broader view of a larger component or a more fine-grained control over a smaller one. On the bottom (**D**), several cases are shown with forced values for metrics and/or elastic strategies. This way, causal inference can improve the systems model to deal with future situations that are not yet known, providing a basis to deal with unexpected or unforeseen scenarios.

4. Conclusions

This article has presented a novel discussion of fundamental aspects that require agreement among the community to enable the development of Distributed Computing

Continuum Systems. This work aims at deviating from the mainstream topics on Distributed Computing Continuum, which usually tackles one of these topics by solving a specific issue, by focusing on the system's representation, its model, its lifelong learning framework, and finally, its business model. This research presents some techniques and concepts that can help address the issues of each of the discussed characteristics, and we compare them with some of the current trends for Distributed Computing Continuum Systems. Further, we show some relations among the presented techniques that can use their synergies to enable future Distributed Computing Continuum Systems.

References

1. Beckman, P.; Dongarra, J.; Ferrier, N.; Fox, G.; Moore, T.; Reed, D.; Beck, M. Harnessing the computing continuum for programming our world. In *Fog Computing*; Zomaya, A., Abbas, A., Khan, S., Eds.; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2020; pp. 215–230. [CrossRef]
2. Morichetta, A.; Casamayor Pujol, V.; Dustdar, S. A roadmap on learning and reasoning for distributed computing continuum ecosystems. In Proceedings of the IEEE International Conference on Edge Computing (EDGE), Chicago, IL, USA, 5–10 September 2021; Institute of Electrical and Electronics Engineers: New York, NY, USA, 2021; pp. 25–31. [CrossRef]
3. Costa, B.; Bachiega, J., Jr.; de Carvalho, L.R.; Araujo, A.P. Orchestration in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–34. [CrossRef]
4. Dustdar, S.; Casamayor Pujol, V.; Donta, P.K. On distributed computing continuum systems. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 4092–4105. [CrossRef]
5. Yu, S.; Chen, X.; Zhou, Z.; Gong, X.; Wu, D. When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network. *IEEE Internet Things J.* **2021**, *8*, 2238–2251. [CrossRef]
6. Xia, X.; Chen, F.; He, Q.; Grundy, J.C.; Abdelrazek, M.; Jin, H. Cost-Effective App Data Distribution in Edge Computing. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 31–44. [CrossRef]
7. Ullah, A.; Dagdeviren, H.; Ariyattu, R.C.; DesLauriers, J.; Kiss, T.; Bowden, J. MiCADO-Edge: Towards an Application-level Orchestrator for the Cloud-to-Edge Computing Continuum. *J. Grid Comput.* **2021**, *19*, 47. [CrossRef]
8. Hastbacka, D.; Halme, J.; Barna, L.; Hoikka, H.; Pettinen, H.; Larranaga, M.; Bjorkbom, M.; Mesia, H.; Jaatinen, A.; Elo, M. Dynamic Edge and Cloud Service Integration for Industrial IoT and Production Monitoring Applications of Industrial Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2022**, *18*, 498–508. [CrossRef]
9. Pusztai, T.; Nastic, S.; Morichetta, A.; Casamayor Pujol, V.; Dustdar, S.; Ding, X.; Vij, D.; Xiong, Y. A Novel Middleware for Efficiently Implementing Complex Cloud-Native SLOs. In Proceedings of the 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 5–10 September 2021.
10. Gheibi, O.; Weyns, D. Lifelong self-adaptation: Self-adaptation meets lifelong machine learning. In Proceedings of the SEAMS'22—17th Symposium on Software Engineering for Adaptive and Self-Managing Systems, Pittsburgh, PA, USA, 22–24 May 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 1–12. [CrossRef]
11. Kirchhoff, M.; Parr, T.; Palacios, E.; Friston, K.; Kiverstein, J. The Markov blankets of life: Autonomy, active inference and the free energy principle. *J. R. Soc. Interface* **2018**, *15*, 20170792. [CrossRef]
12. Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1988.
13. Forti, S.; Bisicchia, G.; Brogi, A. Declarative continuous reasoning in the cloud-IoT continuum. *J. Log. Comput.* **2022**, *32*, 206–232. [CrossRef]
14. Rihan, M.; Elwekeil, M.; Yang, Y.; Huang, L.; Xu, C.; Selim, M.M. Deep-VFog: When Artificial Intelligence Meets Fog Computing in V2X. *IEEE Syst. J.* **2021**, *15*, 3492–3505. [CrossRef]
15. Esfahani, N.; Malek, S. Uncertainty in self-adaptive software systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7475 LNCS, pp. 214–238. ISBN: 9783642358128. [CrossRef]
16. Pearl, J.; Mackenzie, D. *The Book of Why: The New Science of Cause and Effect*, 1st ed.; Basic Books, Inc.: New York, NY, USA, 2018.

17. Casamayor Pujol, V.; Raith, P.; Dustdar, S. Towards a new paradigm for managing computing continuum applications. In Proceedings of the IEEE 3rd International Conference on Cognitive Machine Intelligence, CogMI 2021, Virtual, 13–15 December 2021; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2021; pp. 180–188. [CrossRef]
18. Friston, K.; Kilner, J.; Harrison, L. A free energy principle for the brain. *J. Physiol. Paris* **2006**, *100*, 70–87. [CrossRef]
19. Palacios, E.R.; Razi, A.; Parr, T.; Kirchhoff, M.; Friston, K. On Markov blankets and hierarchical self-organisation. *J. Theor. Biol.* **2020**, *486*, 110089. [CrossRef]
20. Smith, R.; Friston, K.J.; Whyte, C.J. A step-by-step tutorial on active inference and its application to empirical data. *J. Math. Psychol.* **2022**, *107*, 102632. [CrossRef]
21. Dustdar, S.; Guo, Y.; Satzger, B.; Truong, H.L. Principles of elastic processes. *IEEE Internet Comput.* **2011**, *15*, 66–71. [CrossRef]
22. Nastic, S.; Morichetta, A.; Pusztai, T.; Dustdar, S.; Ding, X.; Vij, D.; Xiong, Y. SLOC: Service level objectives for next generation cloud computing. *IEEE Internet Comput.* **2020**, *24*, 39–50. [CrossRef]
23. Stafford, V. Zero trust architecture. *NIST Spec. Publ.* **2020**, *800*, 207.
24. Truong, H.L.; Dustdar, S.; Leymann, F. Towards the Realization of Multi-dimensional Elasticity for Distributed Cloud Systems. *Procedia Comput. Sci.* **2016**, *97*, 14–23. [CrossRef]
25. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020.
26. LiWang, M.; Gao, Z.; Hosseinalipour, S.; Dai, H. Multi-Task Offloading over Vehicular Clouds under Graph-based Representation. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Virtual, 7–11 June 2020; pp. 1–7. [CrossRef]
27. Zhao, Z.; Verma, G.; Rao, C.; Swami, A.; Segarra, S. Distributed scheduling using graph neural networks. In Proceedings of the ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings, Toronto, ON, Canada, 6–11 June 2021; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2021; pp. 4720–4724. [CrossRef]
28. Yu, Z.; Hu, J.; Min, G.; Wang, Z.; Miao, W.; Li, S. Privacy-Preserving Federated Deep Learning for Cooperative Hierarchical Caching in Fog Computing. *IEEE Internet Things J.* **2022**, *9*, 22246–22255. [CrossRef]
29. Zhang, K.; Cao, J.; Zhang, Y. Adaptive Digital Twin and Multi-agent Deep Reinforcement Learning for Vehicular Edge Computing and Networks. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1405–1413. [CrossRef]
30. Sheng, S.; Chen, P.; Chen, Z.; Wu, L.; Yao, Y. Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing. *Sensors* **2021**, *21*, 1666. [CrossRef]
31. Xia, X.; Chen, F.; He, Q.; Cui, G.; Grundy, J.C.; Abdelrazek, M.; Xu, X.; Jin, H. Data, User and Power Allocations for Caching in Multi-Access Edge Computing. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 1144–1155. [CrossRef]
32. Tadakamalla, V.; Menasce, D. Autonomic Elasticity Control for Multi-server Queues under Generic Workload Surges in Cloud Environments. *IEEE Trans. Cloud Comput.* **2020**, *10*, 984–995. [CrossRef]
33. Guo, S.; Wu, D.; Zhang, H.; Yuan, D. Queueing Network Model and Average Delay Analysis for Mobile Edge Computing. In Proceedings of the 2018 International Conference on Computing, Networking and Communications, ICNC 2018, Maui, HI, USA, 5–8 March 2018; pp. 172–176. ISBN: 9781538636527. [CrossRef]
34. Villari, M.; Fazio, M.; Dustdar, S.; Rana, O.; Ranjan, R. Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Comput.* **2016**, *3*, 76–83. [CrossRef]
35. Gamal, I.; Abdel-Galil, H.; Ghalwash, A. Osmotic Message-Oriented Middleware for Internet of Things. *Computers* **2022**, *11*, 56. [CrossRef]
36. Camara, J.; Muccini, H.; Vaidhyanathan, K. Quantitative verification-aided machine learning: A tandem approach for architecting self-adaptive IoT systems. In Proceedings of the IEEE 17th International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, 16–20 March 2020; pp. 11–22. ISBN: 9781728146591. [CrossRef]
37. Thrun, S. Lifelong Learning Algorithms. In *Learning to Learn*; Thrun, S., Pratt, L., Eds.; Springer US: Boston, MA, USA, 1998; pp. 181–209. [CrossRef]
38. Yang, F.; Yang, C.; Liu, H.; Sun, F. Evaluations of the Gap between Supervised and Reinforcement Lifelong Learning on Robotic Manipulation Tasks. In Proceedings of the 5th Conference on Robot Learning. PMLR, London, UK, 8–11 November 2022; pp. 547–556.
39. Febrinanto, F.G.; Xia, F.; Moore, K.; Thapa, C.; Aggarwal, C. Graph Lifelong Learning: A Survey. *IEEE Comput. Intell. Mag.* **2023**, *18*, 32–51. [CrossRef]
40. Aguilera, M.; Millidge, B.; Tschantz, A.; Buckley, C.L. How particular is the physics of the free energy principle? *Phys. Life Rev.* **2021**, *40*, 24–50. [CrossRef]
41. Raja, V.; Valluri, D.; Baggs, E.; Chemero, A.; Anderson, M.L. The Markov blanket trick: On the scope of the free energy principle and active inference. *Phys. Life Rev.* **2021**, *39*, 49–72. [CrossRef] [PubMed]
42. Da Costa, L.; Parr, T.; Sajid, N.; Veselic, S.; Neacsu, V.; Friston, K. Active inference on discrete state-spaces: A synthesis. *J. Math. Psychol.* **2020**, *99*, 102447. [CrossRef]