Health of Smart Ecosystems

Noura El Moussa
Davide Molinelli
Mauro Pezzè
USI Università della Svizzera italiana
SIT Schaffhausen Institute of Technology
Lugano, Schaffhausen, Switzerland
noura.el.moussa@usi.ch|dm@sit.org|mp@sit.org

Martin Tappler
Graz University of Technology
Silicon Austria Labs, TU Graz - SAL DES Lab
Graz, Austria
martin.tappler@ist.tugraz.at

ABSTRACT

Software is a core component of *smart ecosystems*, large 'system communities' that emerge from the composition of autonomous, independent, and highly heterogeneous systems, like smart cities, smart grids, smart buildings. The systems that comprise smart ecosystems are not centrally owned, and mutually interact both explicitly and implicitly, leading to unavoidable contradictions and failures. The distinctive characteristics of smart ecosystems challenge software engineers with problems never addressed so far. In this paper we discuss the big challenge of defining a new concept of 'dependability' and new approaches to reveal smart ecosystem failures.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**; • **Computing methodologies → Intelligent agents**; **Cooperation and coordination**; **Multi-agent systems**; *Artificial life*; *Visual analytics*.

KEYWORDS

systems of systems, ultra-large systems, smart ecosystems, ecosystem health, software verification

ACM Reference Format:
Noura El Moussa, Davide Molinelli, Mauro Pezzè, and Martin Tappler. 2021. Health of Smart Ecosystems. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21), August 23–28, 2021, Athens, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3468264.3473137

1 INTRODUCTION

The remarkable advances in machine learning, IoT and socio-economic systems give rise to *Smart EcoSystems* (*SES*), multifaceted systems that emerge from the composition of independently-operated and autonomous systems with smart functionalities, and that evolve over time, as in the case of next-generation smart cities, smart buildings, smart grids, and more generally a future smart planet.

ESEC/FSE '21, August 23–28, 2021, Athens, Greece

ACM ISBN 978-1-4503-8562-6/21/08…$15.00
https://doi.org/10.1145/3468264.3473137

SES properly extend and enrich the important class of systems variously referred to as "virtual systems of systems" (SoS), "ultra-large scale systems" (ULS), "large-scale complex IT systems" (LSCITS), "cyber-physical systems" (CPS). The autonomous smart systems that comprise *SES* raise new challenges that derive from (i) the heterogeneity and autonomy of the comprising systems, (ii) the complexity of their inter-dependencies, and (iii) the uncertainty with respect to the environments wherein these systems operate.

Maier highlights the absence of central control of virtual SoS [6], Northtop et al. emphasize the challenges of large-scale ULS [9], Sommerville et al. highlight the complexity of LSCITS [11], Cioroaica et al. study CPS [3]. None of them deals with the new challenges of verifying autonomous smart systems.

The recent body of work on verification and validation of machine learning and artificial intelligent systems [13] focuses on single smart systems, largely ignoring the new challenges that emerge from the autonomous composition of independent smart systems. In this paper we discuss the big challenge of defining a new concept of 'dependability' and new approaches to reveal *SES* failing frameworks. We charaterize the relevant class of *smart ecosystems* (*SES*), as *interacting systems of systems that involve heterogeneous smart systems with autonomous behavior*, discuss the distinctive characteristics of *SES*, identify the new challenges of verifying *SES*, propose a novel concept of *ecosystem health* that captures the emerging quality aspects of *SES*, and indicate future research directions towards effective verification of *SES*.

2 SMART ECOSYSTEM

Smart ecosystems (*SES*) emerge from autonomous and independently operating systems of heterogeneous nature (software systems, cyber-physical systems, and people, with an increasingly prominent role of machine learning and computing intelligence) that act and interact in a shared environment. Smart cities, smart grids, smart buildings and peer-to-peer ride-sharing systems are intuitive examples of *SES*.

In this paper, we refer to peer-to-peer ride-sharing *SES*, inspired from Uber, Grab and Blablacar, as a running example of *SES*. Ride-sharing *SES* emerge from the combination of software systems, like ride-managers and payment service systems, physical and cyber-physical systems, like cars, GPS systems and traffic alert systems, and people, like passengers and drivers. Machine learning and computing intelligence play an increasingly prominent role with smart cars, smart ride-managers and smart traffic alert systems. The distinctive characteristics of *SES* are: (a) *autonomy*, (b) *implicit interactions*, and (c) *intrinsic contradictions.*

Autonomy. The systems comprising a *SES* are remarkably heterogeneous, and are characterized by a high degree of operational autonomy. For example, ride-managers, that is, the infrastructures that mediate between ride requests and offers, payment systems, cars, traffic alert and GPS systems, are owned by different entities, and operate autonomously according to their own goals, serving passengers and drivers, who become integral part systems with their autonomous behavior and decisions.

Implicit interactions. The systems in a *SES* interact implicitly and unconsciously through shared resources and hidden actions.

Implicit interactions cannot be designed and tested, but only observed when emerging in *SES*, and can lead to unforeseen scenarios.

For example, an uncontrollable event, like the termination of a sport event or a failure in an underground station, may trigger implicitly interactions due to a sudden increase of request for rides in a specific area. The ride-manager may broadcast a call for drivers. Some drivers may move from neighboring areas to the interested area, causing a shortage of drivers in other areas, leading to new calls for drivers. The events implicitly interact with fares that the ride-manager adapts according to the pending requests and the availability of drivers in different areas, to implicitly control the amount of drivers who move to areas with high fares When drivers served the sudden peak of ride requests, the *SES* goes back to a commonly experienced behavior (*Peak request* scenario).

However, a broadcast about a long-lasting underground station failure, or the breaking news of a sudden flash mob that jeopardize the traffic in some areas, may lead to an anomalous distribution of drivers in the areas affected by the failing events and in the nearby areas, and may implicitly trigger both a sudden increase of fares and an escalation of delays in rides. *'Greedy drivers'* may decline invitations waiting for further increases of fares[1]. While in the many common situations observed so far in the field, ride-sharing *SES* self-balance, an unforeseen event like the announcement of a long-lasting major underground failure or a gigantic flash mob can trigger an amount of declines of 'greedy drivers', an enormous increase in fares and delays, a massive amount of potential passengers giving up, and ultimately a crash of the *SES* with too many unsatisfied requests (*Greedy crash* scenario).

In both scenarios, unknown events out of the control of the *SES* trigger implicit interactions that perturbate the *SES*. However, the Peak request scenario perturbs the *SES* with no major pervasive consequences, while the Greedy crash scenario leads to a disaster situation. All systems comprising the *SES* behave as specified and expected: the ride-manager calls for drivers in areas with a shortage of availability, and adjusts fares to incentivize drivers. Drivers autonomously decide to move to different areas, decline and accept requests according to maximize their revenue. Passengers give up rides when inconvenient for costs or duration. The situation is triggered by events that the *SES* may not be even aware of, like a radio broadcast, and the disaster is a consequence of many implicit interactions that occur spontaneously in the *SES*.

There is evidence of many analogous cases in *SES* as well as in many other ecosystem of different nature. A notable example is the U.S. equity markets crash of May 6, 2010, a well-known *Flash Crash*. The exceptional stock market crash was triggered by a single event that was executed with uncommon urgency, and led to a complex pattern of implicit interactions, causing a temporary loss of about 800 billion US dollars of market value. The cause of the Flash Crash cannot be attributed to a software bug, but rather to a combination of interactions that implicitly led to an unforeseeable scenario [4].

[1]The phenomenon of declining requests looking for increase of fares is a hypothesis of experts eu.usatoday.com),

Intrinsic Contradictions. The autonomy of the systems comprising a *SES*, the lack of shared control and goals, and the presence of implicit interactions lead to implicit contradictions that can strongly affect the stability of the *SES*. Indeed, although *SES* aim to improve collective wellness, the systems interact autonomously to gain their own advantages and compete on common limited resources to reach their sometime contradicting goals.

The *greedy drivers* example well describes the implicit contradictions among the systems that comprise a ride-sharing *SES*. The different goals of traffic alert, pollution control, drivers and passengers are other excellent examples of implicit contradictions in goals and behaviors in an extended ride-sharing *SES*. The traffic alert systems aim to divert traffic away from congested areas, the pollution control systems aim to reduce traffic in pollution-critical areas, drivers aim to get the most profitable routes, and passengers look for the cheapest and fastest routes. Moving traffic from congested areas to solve a traffic jam may increase traffic in critical pollution area, may slow down some drivers and worsen fares. Reducing and slowing down traffic in critical pollution areas may further complicate the interactions.

The implicit contradictions in *SES* may lead to selfish decisions that can degrade the quality of the *SES* both locally and globally, threatening the availability and distribution of the resources. For instance, Thai et al. [12] observe that the selfish intensive use of routing apps on road usage, such as Google Maps, Waze, INRIX, or Apple Maps, all of which take into account only the needs of the individual user by suggesting alternative routes to avoid traffic jams, cause a sharp increase in traffic on local roads without benefiting highway congestion.

3 SMART ECOSYSTEM HEALTH

The *Greedy crash* scenario well exemplifies the new failing framework of *SES*: An intolerable persistent increase of fares that leads to a generalized cancelation of ride requests with most drivers idle, no passengers served and no revenues whatsoever, that is, a main degenerations of the behavior of the whole *SES*, as a consequence of unforeseen *implicit interactions* of *autonomous* systems that behave correctly with respect to their specifications.

The *implicit interactions* and *intrinsic contradictions* of *SES* challenge the classic notions of dependability and failures, defined as deviations of the observed from the specified behavior [1], being it impossible to specify the correctness of a *SES* in the presence of contradicting requirements of the autonomous systems.

In this paper we discuss the big challenge of defining a new concept of 'dependability' and new approaches to reveal SES *failing frameworks.*

Sommerville et al. [11] already notice that the classic notion of failure is not applicable, and observe that LSCITS present 'wicked' problems, which constantly change. In this paper, we propose a novel definition of the expected behavior of a *SES* that we call

SES *health*, borrowing the term from software ecosystems [7] that are defined as software systems composed of actors that provide services according to well-defined system requirements and are thus radically different from *SES*, as defined in this paper.

Getting inspiration from popular definitions of business [5] and natural [10] ecosystems, which identify three dimensions of healthiness, we define *smart* SES *health* along three non-orthogonal dimensions, *hardiness*, *consistency*, and *resilience*, borrowing the terms from existential and humanistic psychology. Intuitively, *hardiness* indicates the strength of a *SES* in terms of amount and quality of the provided services, by referring to the existential metaphor of the strength of human beings; *consistency* captures the ability of the *SES* to exhibit a stable behavior also in the presence of sudden and unpredictable events, and follow predictable patterns, by referring to the metaphor of the human personality, *resilience* captures the ability of the *SES* to react to unavoidable crisis and go back to an acceptable behavior. We quantify and monitor the health of *SES* by referring to domain-specific and quantifiable *indicators* that capture the distinctive aspects of these dimensions.

We identify two kinds of indicators, *global* SES *health indicators* and *specific system indicators*. Global health indicators capture the quality of *SES* as a whole, and are independent from the contradicting requirements of the individual systems in a *SES*. Specific indicators refer to subsets of systems, and are subject to different interpretations driven by the contradicting requirements of the individual systems. Indicators may vary across different types of *SES* and among *SES* of the same type.

Examples of global indicators of ride-sharing *SES* are requested rides, completed rides, percentage of satisfied ride requests (ration of completed over requested rides), ride tardiness (difference between expected and actual ride time), service slack time (interval between request and start of a ride), ride safety (amount of violations of the traffic regulations per distance and time).

hardiness, *consistency* and *resilience* emerge as different combination of *SES* health indicators: The percentage of satisfied ride requests is a key component of the *hardiness*, the variation of ride tardiness and service slack time quantify the *consistency* of the *SES*, the fluctuation of the percentage of satisfied ride requests and tardiness refer to the *resilience* of a share-ride *SES*.

Such indicators capture the quality of a ride-sharing *SES* independently from passengers, drivers, and cloud mediators: the more the satisfied rides are, the lower the tardiness, the service slack time and the safety are, the better the health of the ride-sharing *SES* is.

Example of specific indicators of ride-sharing *SES* are ride cost, ride price for passenger, driver compensation, mediator remuneration, whose overall expectations radically differ and contradict each other. Other examples of specific indicators are the amount of available drivers and ride requests in the same and neighboring areas, that raise different expectations due to their impact on fares and service.

While the correctness of different systems are directly related to the value of the indicators, the health of the whole *SES* depends on complex patterns of (implicit) interactions among systems, interactions that we capture as relations among the *SES* indicators. The values of the *SES* indicators vary over time depending on the autonomous behavior of the individual systems, and may reflect natural adaptations of the *SES* to evolving situations as well as *SES* health failures, that is, major degradations of the *SES* as whole.

We define a *SES health failure* as an unacceptable degradation of some global *SES* indicators, for example, an exceptional drop of satisfied ride requests, in the case of the *greedy crash*, or the exceptional loss of market value, in the case of the *flash crash*. While degradations of *SES* indicators can be identified by looking at the variation of the values of the indicators over time, acceptable degradations can be defined only with respect to the behavior of the autonomous systems.

For example a sudden decrease of the percentage of satisfied requests, and an increase of ride tardiness and service slack time (degradation of some *SES* indicators) may be a temporary undesirable but unavoidable transient situation, if they happen jointly with a quick increase of ride requests, a fast reduction of available drivers, and an increase of ride prices (specific system indicators), for instance due to the sudden termination of a popular event that generates a rapid increase of requests, and temporary exhausts the availability of drivers, soon compensated by calls for drivers in areas with high fares: the *Peak request* scenario. However, a similar trend in the degradation of the same *SES* indicators may be a *SES* failure, if experienced with a quick reduction of available drivers and an increase of fares with an initially constant and then rapidly decreasing amount of requests: the *Greedy crash* scenario.

We define a SES *fault* as a complex pattern of (implicit) interactions that lead to an *SES* failure, for instance the complex patterns of events that lead to *Greedy crashes* in a ride-sharing *SES*, or the complex pattern of events that that lead to a flash crash in an equity market. We define a *trigger* as the (single) event that ignite a *SES* fault, like the radio broadcast about the duration of the underground failure, or the single huge block sale that was executed with uncommon urgency, and led to a flash crash.

SES emerge and evolve over time when systems seamlessly enter, exit and interact in the *SES*. While we can verify and test systems before deployment to identify and remove faults, it it impossible to fully verify and test *SES* before they emerge and evolve in the field. A big challenge of verifying *SES* in the field, is to precisely identify *SES* faults before *SES* health failures, the closest the possible to the triggers, that is, distinguishing *SES* faults from atypical interaction patterns due to anomalous events, before *SES* fails, for instance, distinguishing *Greedy crash* from *Peak request* scenarios in ride-sharing *SES*, or scenarios due market perturbations from scenarios that lead to flash crashes in equity markets.

4 VERIFYING THE HEALTH OF *SES*

The new concepts of *SES* health and failure present new challenges and requires new approaches for verifying *SES*. In this section, we propose a new approach to verify the healthiness of *SES*, illustrate it by referring to the ride sharing scenarios, and present the results of a preliminary investigation that indicate the feasibility of the proposed approach.

Our goal is to predict *SES* failures before *SES* fails, by revealing faults, that is, predict unacceptable degradations of *SES* indicators, by revealing pattern of (implicit) interactions that lead to an *SES* failure. Failures triggered by unforeseen events, and due to (implicit)

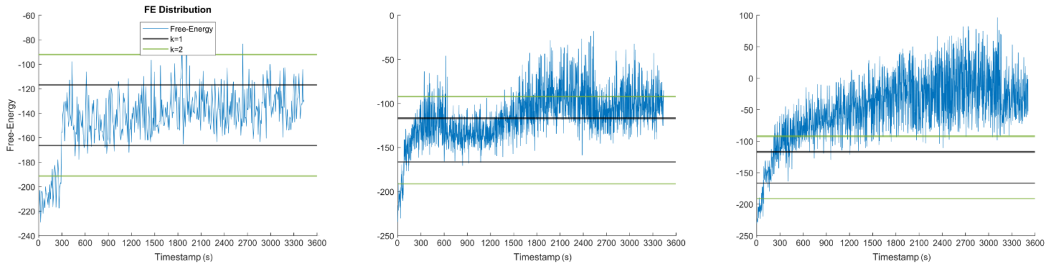

Figure 1: Energy values for consistency in the three exemplar scenarios

interactions that emerge at runtime in evolving *SES*, can be detected only by observing the behavior of *SES* at runtime.

By inspiring from the work of Monni and Pezzè on predicting failures by detecting energy anomalies of system metrics [8], we predict *SES* failures by detecting energy anomalies of global and specific indicators as symptoms of *SES* faults. We compute the energy of *SES* with Restricted Boltzmann Machines (RBMs) [2], bipartite neural networks that infer energy-based models by using an energy function to derive the marginal distribution of the input to approximate the output distribution of the modeled system.

We detect energy-anomalies in two phases: (i) We reveal possible faults as anomalous energy levels computed with an RBM over all indicators, and (ii) diagnose system failures as anomalous energy values computed with RBMs over three (overlapping) subsets of indicators that correspond to *hardiness*, *consistency* and *resilience*.

We got some initial insights with a simple prototype implementation of the ride sharing ecosystem example with passengers, drivers, cars and ride-mediator. [2] We collected 21 global *SES* indicators, and 22 specific indicators for each ride. We collected data sets for normal scenarios with data derived from publicly available Uber data, and data sets for the *Peak request* and *Greedy crash* scenarios.

We selected 33 indicators for *hardiness*, 35 for *consistency*, and 28 for *resilience*, and computed the energy values with respect to the three subsets in the different scenarios. Figure 1 reports the free energy for *consistency* computed for 1 hour runs for the *Normal*, *Peak request* and *Greedy crash* scenarios. The horizontal bars in the diagrams delimit the range of normal values for the free energy, thresholds that we computed during normal execution.

The diagrams indicate that the free energy clearly distinguish *Normal*, anomalous (*Peak request*) and failure prone (*Greedy crash*) scenarios: the free energy remains within, moves temporarily out of, and steadily remains out of the normal range, respectively. The energy values for *hardiness* and *resilience* that we not report for the lack of space present similar characteristics.

5 CONCLUDING REMARKS

In this paper we (i) highlight the distinctive characteristics of *SES*, (ii) discuss the new challenges of detecting *SES* failures that emerge due to unforeseen conditions, (iii) present the new concept of *SES* health to capture the quality of *SES*, (iv) articulate *SES* health into three main dimensions, (v) propose an approach to reveal health anomalies and detect *SES* faults, (vi) present the results of a preliminary experiment that suggest that it is worthwhile to further investigate the research direction drafted in this paper.

Early detecting faults opens new challenges related to preventing failures and discovering triggers, to avoid mayor disasters and design mechanisms to limit the impact of triggers.

[2]The source code of the simulator is publicly available at the linked repository.

REFERENCES

[1] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.* 1, 1 (2004), 11–33. https://doi.org/10.1109/TDSC.2004.2

[2] Miguel Á. Carreira-Perpiñán and Geoffrey E. Hinton. 2005. On Contrastive Divergence Learning. In *Proc. International Workshop on Artificial Intelligence and Statistics*. The Society for Artificial Intelligence and Statistics.

[3] Emilia Cioroaica, Thomas Kuhn, and Thomas Bauer. 2018. Prototyping Automotive Smart Ecosystems. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN, Luxembourg, June 25-28, 2018*. IEEE Computer Society, 255–262. https://doi.org/10.1109/DSN-W.2018.00072

[4] United States. Commodity Futures Trading Commission, United States. Securities, and Exchange Commission. 2010. *Preliminary Findings Regarding the Market Events of May 6, 2010: Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues*. U.S. Commodity Futures Trading Commission. https://books.google.ch/books?id=aCaSnQAACAAJ

[5] Marco Iansiti and Roy Levien. 2004. Strategy as ecology. *Harvard business review* 82, 3 (2004), 68–78.

[6] Mark W. Maier. 1998. Architecting principles for systems-of-systems. *Systems Engineering* 1, 4 (1998), 267–284. https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D

[7] Konstantinos Manikas and Klaus Marius Hansen. 2013. Reviewing the Health of Software Ecosystems - A Conceptual Framework Proposal. In *Proceedings of the 5th International Workshop on Software Ecosystems, Potsdam, Germany, June 11, 2013 (CEUR Workshop Proceedings, Vol. 987)*, Carina Frota Alves, Geir Kjetil Hanssen, Jan Bosch, and Slinger Jansen (Eds.). CEUR-WS.org, 33–44. http://ceur-ws.org/Vol-987/3.pdf

[8] Cristina Monni and Mauro Pezzè. 2019. Energy-based anomaly detection a new perspective for predicting software failures. In *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results, ICSE (NIER) 2019, Montreal, QC, Canada, May 29-31, 2019*. IEEE / ACM, 69–72. https://doi.org/10.1109/ICSE-NIER.2019.00026

[9] L. Northrop, Peter Feiler, R.P. Gabriel, John Goodenough, Rick Linger, Thomas Longstaff, Rick Kazman, M. Klein, Douglas Schmidt, Kevin Sullivan, and Kurt Wallnau. 2006. *Ultra-Large-Scale Systems - The Software Challenge of the Future*. Technical Report. Software Engineering Institute – Carnegie Mellon.

[10] D.J. Rapport, R. Costanza, and A.J. McMichael. 1998. Assessing ecosystem health. *Trends in Ecology & Evolution* 13, 10 (1998), 397 – 402.

[11] Ian Sommerville, Dave Cliff, Radu Calinescu, Justin Keen, Tim Kelly, Marta Z. Kwiatkowska, John A. McDermid, and Richard F. Paige. 2012. Large-scale complex IT systems. *Commun. ACM* 55, 7 (2012), 71–77.

[12] Jerome Thai, Nicolas Laurent-Brouty, and Alexandre M. Bayen. 2016. Negative externalities of GPS-enabled routing applications: A game theoretical approach. In *19th IEEE International Conference on Intelligent Transportation Systems, ITSC, Rio de Janeiro, Brazil, 2016*. IEEE, 595–601. https://doi.org/10.1109/ITSC.2016.7795614

[13] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering, ICSE, Gothenburg, Sweden, 2018*, Michel Chaudron, Ivica Crnkovic, Marsha Chechik, and Mark Harman (Eds.). ACM, 303–314. https://doi.org/10.1145/3180155.3180220