Detecting emergence in engineered systems: A literature review and synthesis approach

(Detecting Emergence in Engineered Complex Systems; a Literature Review and Synthesis Approach)

Haugen, Rune Andre[1]; Skeie, Nils-Olav[2]; Muller, Gerrit[1]; Syverud, Elisabet[1]
[1]Department of Science and Industry systems - University of South-Eastern Norway
[2]Department of Electrical engineering, Information Technology and Cybernetics - University of South-Eastern Norway

This is the peer reviewed version of the following article:

Haugen, R. A., Skeie, N.-O., Muller, G. & Syverud, E. (2023). Detecting emergence in engineered systems: A literature review and synthesis approach. Systems Engineering. https://doi.org/https://doi.org/10.1002/sys.21660

which has been published in final form at [Link to final article using the DOI]. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions. This article may not be enhanced, enriched or otherwise transformed into a derivative work, without express permission from Wiley or by statutory rights under applicable legislation. Copyright notices must not be removed, obscured or modified. The article must be linked to Wiley's version of record on Wiley Online Library and any embedding, framing or otherwise making available the article or pages thereof by third parties from platforms, services and websites other than Wiley Online Library must be prohibited.

© 2023 Wiley Periodicals LLC.

Detecting Emergence in Engineered Complex Systems; a Literature Review and Synthesis Approach

Rune André Haugen
University of South-Eastern Norway
rune.a.haugen@usn.no

Elisabet Syverud
University of South-Eastern Norway
elisabet.syverud@usn.no

Gerrit Jan Muller
University of South-Eastern Norway
gerrit.muller@usn.no

Nils-Olav Skeie
University of South-Eastern Norway
nils-olav.skeie@usn.no

Abstract. Complex systems are prone to emergent behavior, and the industry has a need to establish better practices to detect inherent emergent behavior in engineered complex systems. In this literature review paper, we have investigated the phenomenon of emergent behavior, which philosophers and researchers have debated throughout history, tracing all the way back to the time of the well-known Greek philosopher Aristotle (384-322 B.C.). Our focus was on how the literature describes this phenomenon, as well as how to detect it. Emergence is in general explained as behavior emerging at macro level that cannot be traced back to the micro level. We found three main directions in existing literature on emergence, being: 1) unpredictable form only, 2) predictable form only, and 3) both predictable and unpredictable forms. Emergence is categorized in four main categories, being: 1) simple, 2) weak, 3) strong, and 4) spooky, and relates to the four complexity categories: 1) simple, 2) complicated, 3) complex, and 4) chaotic. We found that simple emergence is intuitive and can be detected first-hand in a system model, while weak emergence is non-intuitive requiring simulation and analysis for detection. We can then update the system models to reflect our new system behavior understanding. We found three main approaches for detecting emergence, being 1) test coverage, 2) manual vs automatic task balance, and 3) modeling & simulation. There are many examples of methods for detecting predictable emergence (simple and weak), but no research showing promising results when it comes to dealing with unpredictable emergence (strong and spooky). We believe some unpredictable (strong) emergence can be shifted towards predictable (weak) through increased system knowledge for the observer.

Keywords — Automation, Complexity, Complex Systems, Emergence, Emergent Behavior, Integration

Introduction

Background

Modern product development consists of state-of-the-art technology, which drives the complexity and makes the system prone to emergent behavior. We observe that industries in the defense, space, and aerospace domains, driven by stringent product requirements and long lifetime hold such complexity. These industries search for methods to detect emergence.[1]

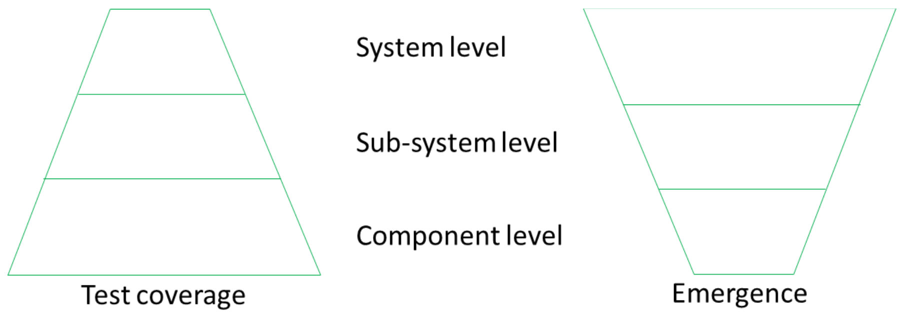

Figure 1: Principal sketch of the mismatch in test coverage and emergence

The system level testing in the industry typically lacks coverage due to the cost of testing realistically at this level. Real-system tests are rare. Close to real-system hardware test arenas are often used for real-time simulation, but even these are used to a limited degree because of the resource situation (people and money) and project schedules. *Figure* 1 shows a principal sketch of the likely mismatch in typical test coverage at different system levels compared to the estimated amount of emergence at different system levels, based on our experience. This lack in test coverage often leads to emergent behavior being discovered late in the development process or not discovered until the operational phase of the product.[2] Kjeldaas et al.[2] indicate that the product developers check only a fraction of the test data. Therefore, future product developments do not benefit from the knowledge in the unchecked data.

While the system tests themselves are expensive and time consuming, the checks are often manual, time consuming, and require system expertise. The consequence is that unforeseen system behavior remains undetected.[2] The challenge is then how to improve the system integration and test process. How can we detect emergent behavior in the system under test that development projects in the industry have failed to do in the past?

Philosophers and researchers have been debating the phenomenon of emergent behavior throughout history. It traces all the way back to the time of the well-known Greek philosopher Aristotle (384-322 B.C.). The famous quote “The whole is greater than the sum of its parts” is related to his quote “The whole is something besides the parts”.[3] One plus one does not equal two, but more due to interaction effects creating new behavior at higher system levels.

We want to conduct a literature review to find typical descriptions of emergence, as well as means to detect emergent behavior. The review will focus on the emergence phenomenon: how it is described, and what methods are used to detect it.

Figure 2 visualizes the four areas of interest for our research case, also illustrating their overlapping areas. The key area to explore will be the area common to all four (gray area). Emergence is the main theme. Complexity relates to emergence in the way that higher emergence categorized behavior typically arises in higher complexity categories, requiring different measures to detect, if at all possible. Automation is important to make us capable of testing and analyzing sufficiently to detect most emergent behavior of the system, and industrial companies typically have a huge potential in shifting the man vs machine task balance.[4] *System integration & test* is the product development domain where we focus our efforts in detecting emergence, which should start at the beginning of the project.

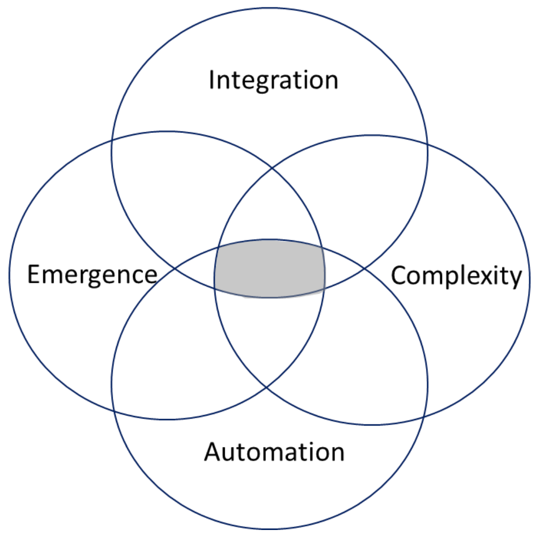

Figure 2: Principal sketch of areas of interest and their overlaps

The *system architecture & design* and *system integration & test* activities should make sure that most system errors- and undesired behaviors are detected. It is essential to balance the time used in these two above-mentioned activities to balance with regards to process efficiency, and how much time we spend on Systems Engineering (SE) effort in total. According to Honour,[5] 15% of project cost spent on SE effort is optimal in the sense of keeping the actual cost- and schedule according to plan.

The scope of our review is to find strategies and methods that make it possible for the industry to detect emergence, where the industry struggles with detecting it today. The literature may help us see which methods are used in relevant engineered complex systems to detect emergence.

Contributions of the review paper

The contributions from this review will include:

- Emergence description and categorization. How the literature explains and categorizes emergence.
- Emergence relation to complexity. How types of emergence appear in types of complexity, and how this can depend on the observer.
- Identification of practical methods to detect emergence.
- Synthesis on the literature on emergence in engineered complex systems.

Research questions

The literature review will answer the following research questions:

- Main research question (RQ1): Where is the research in the area of detecting emergence in engineered complex systems?
- Sub-research question 1 (RQ2): How does the literature explain and categorize emergence?
- Sub-research question 2 (RQ3): What approaches and methods do we see in the literature for detecting emergence?

Scope of survey

The remainder of this paper is structured as follows: Section 2 presents methodology used for this literature review. Section 3 provides an overview of literature when it comes to explaining and detecting the phenomenon of emergence, looking at differences and similarities in taxonomies and methods. Section 4 provides a discussion of the reviewed literature addressing the research questions. Section 5 concludes the paper, provides gained knowledge to detect emergence, defines limitations of the study, and proposes future research.

Methods

The literature review type we will conduct is called narrative synthesis, which is part of the mixed methods review family.[6] We want to get an overview of the research in the area of interest and how we can take advantage of this research using critical reflection in the synthesis process.[6]

We want to summarize the substance of the most relevant literature on emergence, specifically descriptions and methods for detection of this particular phenomenon. It is of interest to investigate and debate the methodologies used, to see if there are any connections between methodologies used and the successful or unsuccessful outcomes. It is also important to look at the practical methods to estimate the relevance for our ongoing research. This is normal for a literature study focusing on research outcomes, and perhaps the most common way to do a literature review according to Randolph.[7]

A goal of the literature review is to establish relevant taxonomies, descriptions, and methods within our field of research that we want to base our further research on. It is important to generalize trends in the literature, or else the level of details would be overwhelming.[7] We will allocate the literature into different directions, discuss these different directions, evaluate them, and conclude on the best way forward. We will discuss the research outcomes found in this literature study in terms of the applicability to our continuing research.

The authors will use a search strategy to crawl the titles, keywords, and abstracts of the databases Web of Science,[8] Scopus,[9] and IEEE Explore[10] for defined search words. In addition, we will try using author keywords and keywords plus. The researchers will use search words combined with Booleans, AND for different elements of the research question, and OR for different relevant synonyms. To include different wording in English and American, like behavior or behaviour, we use truncation (behavio?r). To include all forms of a word (emergent, emergence, emergentism), we use wildcard (emergen*). To restrict the search to specific wording, we use phrases ("Machine Learning").

We will not use proximity operators since librarians at the University of South-Eastern Norway (USN) informed us that limitations should be used carefully, because restricting the search to specific domains could result in important information being missed due to non-optimal handling of these limitations in the searches. We will therefore not use any limitations in our searches but do the screening manually.

Results

Literature Search and Selection

The literature search results can be found in *Table 6* in *Appendix A: Literature Search Results*, which lists the primary sources found through the databases Web of Science,[8] Scopus,[9] and IEEE Explore.[10] The selection process can be seen in *Figure 3,* including three steps of screening:

- First, we scanned through the titles to exclude irrelevant topics, resulting in a down-selection from 645 to 73 records.
- Second, we read the abstracts to remove uninteresting research, resulting in a further down-selection from 73 to 56 records.
- Third, we skimmed through the text of the remaining articles to filter out even more not being sufficiently relevant to our research, resulting in a final down-selection from 56 to 16 records.

The literature search log can be found in *Table 9* in *Appendix B: Literature Search Log,* including more details regarding the searches and findings.

Figure 3 shows the PRISMA flow diagram[11] of the selection process of the primary references found through the database searches. Based on this search and selection process, we found central or pivotal literature within our field of study. This process was not sufficient for us to be able to generalize. Additional literature of interest was found by looking at the reference lists of the selected sources, using the so-called "snowballing effect".[6] This, to include sufficient literature in the field of study to be able to generalize with more confidence. See *Table 7* in *Appendix A: Literature Search Results* for an overview of the secondary sources found through the "snowballing effect" from the primary sources, which we found to support the relevant information in the primary sources. We included 15 new records in this second round of literature selection. To further complement the list of literature, we added a set of different relevant references. See *Table 8* in *Appendix A: Literature Search Results* for an overview of tertiary references based on other channels like colleagues, conferences, and our previous work. These tertiary references completed the survey results to include all relevant literature to our knowledge. We included 25 new record in this third round of literature selection, ending with an overall count of 56 records.

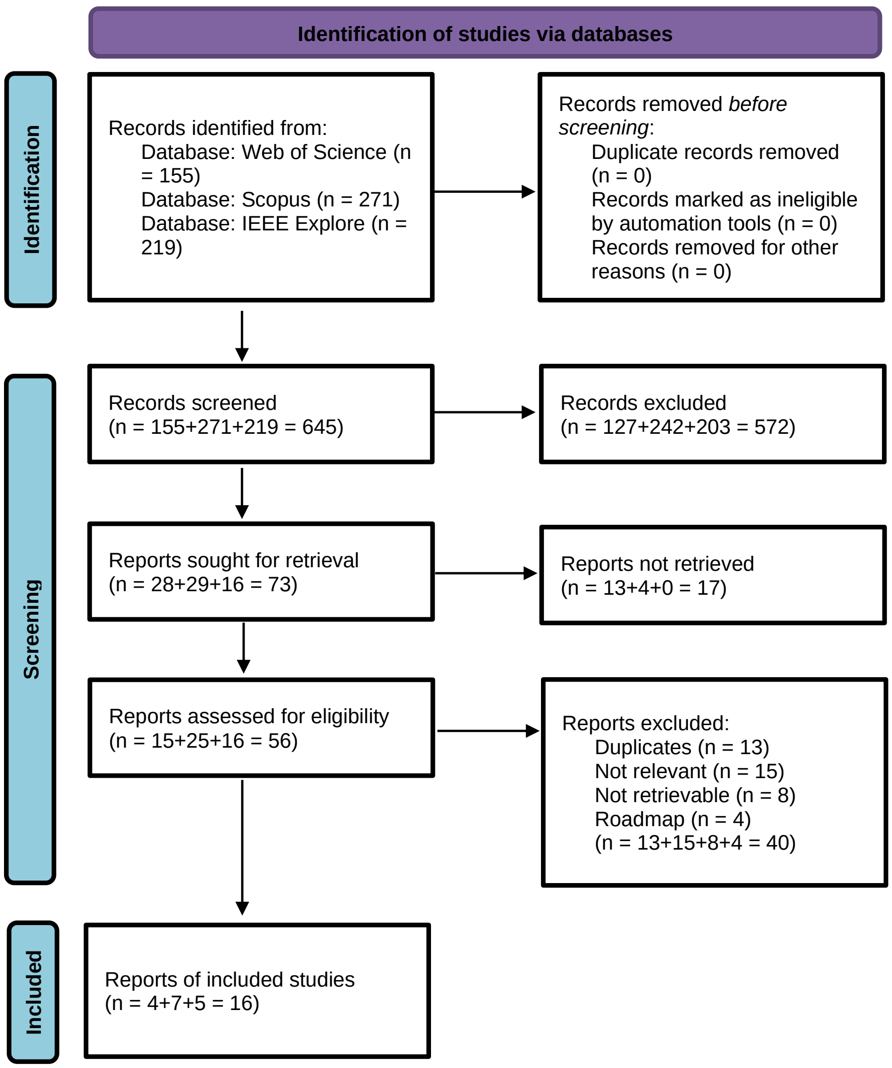

Figure 3: Literature search selection of primary references

Literature Metadata-Analysis and Mapping

We conducted a metadata-analysis of the selected references to ensure their validity by investigating their affiliations, publishing channels, and citations. The metadata-analysis is part of *Table 6*, *Table 7*, and *Table 8* in *Appendix A: Literature Search Results*.

Figure 4 maps the literature selected for further study to see which citation relations there are among these (the color coding is for visualization purposes to see the citations more easily from different authors). Kjeldaas et al.,[2] Mittal et al.,[12] and Szabo and Teo[13] are the sources with the most references to others, containing information from many of the sources of older date. Bedau,[14] Holland,[15] and Mogul[16] are the sources that most of the others are referencing, which we then consider to be of high value.

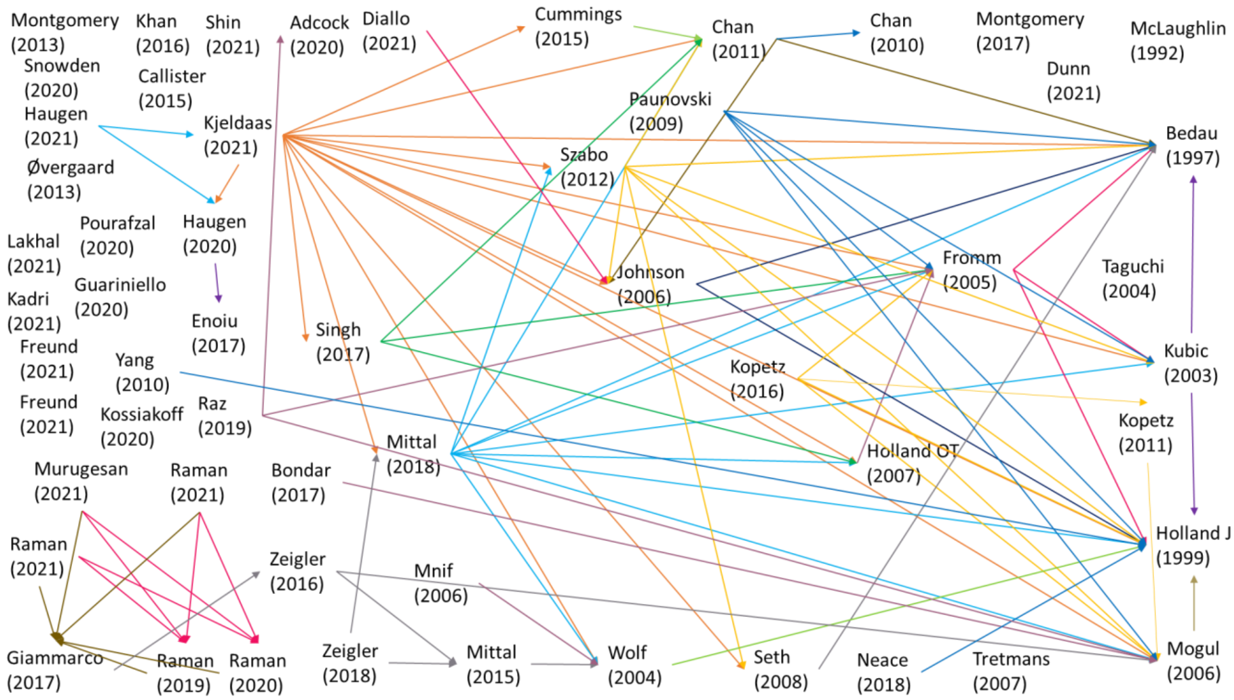

Figure 4: Mapping of relevant literature

Emergence Descriptions

Table 1 shows some representative descriptions of the emergence phenomenon from the literature.

Callister and Andersson[17] use the related term "technical debt", which could result in emergence caused by many small shortcuts during the architecture and design phase.

Table 1: Representative descriptions of the emergence phenomenon from relevant literature

Reference	Description of the Emergence Phenomenon
Holland [15]	"Emergence is above all a product of coupled context-dependent interactions. Technically these interactions, and the resulting system, are nonlinear. The behavior of the overall system cannot be obtained by summing the behaviors of its constituent parts. We can reduce the behavior

Reference	Description of the Emergence Phenomenon
	of the whole to the lawful behavior of its parts, if we take the nonlinear interactions into account.”
Fromm [18]	“A property of a system is emergent, if it is not a property of any fundamental element, and emergence is the appearance of emergent properties and structures on a higher level of organization or complexity.”
Giammarco [19]	“Emergence is a product of the interactions among systems, as well as a product of lack of interactions among systems.”
Holland [20]	“Emergence is that phenomena in which patterns that are observed at a global level arise solely from interactions among lower-level components acting on rules which are executed using only local information without reference to the global pattern.”
Kadri et al. [21]	Sub-systems cooperate in order to carry out a global mission from a set of sub-missions, which each sub-system cannot carry out alone.
Kopetz [22]	“We speak of emergence when the interactions of subsystems give rise to unique global properties at the system level that are not present at the level of the subsystems. Non-linear behaviour of the subsystems, feedback and feed forward mechanisms, and time delays are of relevance for the appearance of emergent properties. Emergent properties are irreducible, holistic, and novel – they disappear when the system is partitioned into its subsystems.”
Wolf and Holvoet [23]	“The collective behaviour is not readily understood from the behaviour of the parts. The collective behaviour is, however, implicitly contained in the behaviour of the parts if they are studied in the context in which they are found. Emergent properties cannot be studied by physically taking a system apart and looking at the parts (=reductionism). They can, however, be studied by looking at each of the parts in the context of the system as a whole.”

Emergence Directions

We see three different directions of emergence, being unpredictable form only, predictable form only, and both predictable and unpredictable forms. *Table 2* shows some representative references from the literature connected to descriptions of the three emergence directions.

Table 2: Representative descriptions of emergence directions from relevant literature

Reference	**Direction of the Emergence Directions**
McLaughlin [24]	Unpredictable form only: Emergence is connected to unpredictability, dismissing predictable forms of emergence by stating that a property is either reducible or emergent. Unpredictability does not equal emergence, since not all that is unpredictable is emergent. We can see physics, chemistry, and biology as sub-systems. When nature integrates these sub-systems into a system, emergent behavior will show. Scientific advances have reduced the domain of emergentism.
Bedau [14], Mnif and Muller-Schloer [25]	Predictable form only: Strong (unpredictable) emergence is possible, but at the same time uncomfortably like magic and getting something out of nothing. Weak (predictable) emergence can explain autonomous behavior at macro level due to micro level dynamics. Complex systems exhibit weak emergence, and emergence can be both foreseen and unforeseen.
Mittal et al. [12], Fromm [18], Holland [20], Mittal and Rainey [26], Seth [27]	Predictable and unpredictable forms: Emergence can be divided into a deterministic and a stochastic region, deterministic being predictable and stochastic being unpredictable. Strong emergence is just outside the bounds of deterministic systems, entering the stochastic region. In this region, with the help of available knowledge, the stochasticity can be controlled. Once new knowledge about the novel behavior is obtained the system now portrays weak emergent behavior since the behavior is not novel anymore.

Kopetz et al.[28] bring another view to emergence. "We typically exploit expected beneficial emergence for intended product performance, while unexpected beneficial emergence can result in the product performing better than that for which it was designed. Expected detrimental emergence typically is allowed weaknesses of a product, while unexpected detrimental emergence could lead to unpredictable system failure".[28] Johnson,[29] as well as Zeigler,[30] supports this distinction between beneficial (positive) and detrimental (negative) emergence. Kopetz et al.[28] and Mittal et al. [12], although agreeing on the direction of emergence, differ in how the observer influences the definition of emergence. Mittal et al. [12]

include the observer in the definition of emergence by looking at how the phenomenon is experienced by the observer. Kopetz et al.[28] on the other hand excludes the observer from the definition of emergence, saying the phenomenon is what it is independent of the observer.

From the Systems Engineering Body of Knowledge (SEBoK)[31] we see that sufficient operational experience is necessary to put us in a position to be able to exploit positive- and avoid negative emergence, and that preventive actions alone are not enough. "True hindsight and understanding comes from building multiple systems of the same type and deploying them, then observing their emergent behavior in operation and the side effects of placing them in their environments. If those observations are done systematically, and the emergence and side effects are distilled and captured in relation to the design of the systems — including the variations in those designs — and made available to the community, then we are in a position to predict and exploit the emergence".[31]

Emergence and Complexity

Emergence and complexity are highly related terms, and this coupling is necessary to understand when researching the emergence phenomenon.

From Kopetz[22]: "We classify a system as complex if we are not in the position to develop a set of models of adequate simplicity – commensurate to the rational capabilities of the human mind – to explain the structure and behaviour of the system. Examples for complex systems are the earth's climate and weather, the global economy, living organisms, and many large computer systems".

From Mittal et al.[12] :"In complex systems, the cause-and-effect relations are only coherent in retrospect and usually do not repeat. Although the behaviour is consistent and explainable within the system, it is not reproducible (otherwise the systems were complicated, not complex)."

Freund[32,33] introduces three dimensions to complexity, being 1) structural complexity, 2) dynamic complexity, and 3) environmental complexity. The combinations of the three dimensions can result in complicated-, structurally complex-, dynamically complex-, environmentally complex-, complex-, and chaotic system.[32,33]

The Cynefin framework[34] proposes ways to deal with different degrees of complexity in a system, and that analysis can help transitioning to a more favorable category.

Mnif and Muller-Schloer[25] states that order depends on subjective decisions or capabilities of the observer, which indicates a similar subjectivity to complexity as Mittal et al.[12] has to emergence.

Emergence Detection

In this section, we look at different approaches and methods to detect emergence, which are summarized in *Table 3, Table 4,* and *Table 5*.

According to Mogul[16]: "We will never be able to solve all emergent misbehavior problems, especially as system complexity increases. However, we can and should be able to recognize recurring patterns of misbehavior, and to learn enough from experience to be able to avoid or repair many of the common patterns."[16]

Table 3: Representative literature highlights on the test coverage approach and methods to detect emergence

References	**Literature Highlights on Emergence Detection Methods**
Haugen and Ghaderi[35]	The design of experiments is an important approach to facilitate detection of emergent behaviors.
Dunn [36], Montgomery [37]	Fractional factorial design of experiments is a method to reduce the scope of testing.
Taguchi et al. [38]	The Taguchi method is a method for systematic testing through orthogonal arrays.
Khan and Jing[39]	Instead of exploring all of the possible system state space, which may grow exponentially, the temporal logic method helps in computing the polarization and momentum for generating the emergent behavior set, and accordingly from which the emergent behaviors can be detected.

Table 4: Representative literature highlights on manual vs automatic task balance approach and methods to detect emergence

References	**Literature Highlights on Emergence Detection Methods**
Haugen and Mansouri[4]	Automating test execution and test result analysis can remove bottlenecks in the test process.
Szabo and Teo[13]	A key challenge is the need for abstractions of the micro and macro levels, which are difficult to achieve in an automated manner, and hence most approaches rely on a post-mortem observation of the simulation by a system expert.

References	Literature Highlights on Emergence Detection Methods
Giammarco[19]	Automated tools with built in simulators become essential for verifying and validating behavior logic in a reasonable amount of time.
Raz et al.[40]	Systems Engineering is the first step towards understanding and controlling emergent behavior, while future methods are expected to rely on Machine Learning methods.
Raman et al.[41-45]	The Measures of Performance (MoP's) and Measures of Effectiveness (MoE's) can be monitored using Machine Learning to look for changes that give or could give raise to emergent behaviors.
Diallo et al.[46]	Suspicious values produced by the statistical debugging process can be manually inspected to determine emergent behavior.
Øvergaard and Muller[47], Enoiu et al.[48]	We can reduce the test time used for manual testing by more than 90% by automating the test procedures that are suitable for automation.

Table 5: Representative literature highlights on modeling & simulation approach and methods to detect emergence

References	Literature Highlights on Emergence Detection Methods
Guariniello et al.[1]	System modeling can give direct insight into the causes of observed, possibly emergent behavior.
Holland[15]	Insight can only be obtained with the help of computer-based exploration.
Bondar et al.[49]	"An accurate and complete system architecture model for a System-of-Systems (SoS) is required to measure the existence, type, and level of emergent behavior of the SoS."
Kossiakoff et al.[50]	In a system behavior model, we can analyze performance at multiple levels (system, sub-system, component).

References	Literature Highlights on Emergence Detection Methods
Neace and Chipkevich[51]	Emergence requirements can be used as a method for governance of emergence in a complex system.
Zeigler et al.[52]	Model Based Systems Engineering (MBSE) can benefit from simulations for early validation of the system design.
Montgomery[53]	Model Based Systems Integration (MBSI) extends the MBSE process by increasing the system integration impact early in the design.
Cummings[54]	A system of systems modeling and simulation framework architecture can provide identification and quantification of emergent behavior.
Paunovski et al.[55]	An emergent phenomenon is only visible at runtime operation and cannot be captured apriori with a model of the system, but the conceptual models can be updated with the gained knowledge through simulations.
Kubic[56]	The sources of emergence can be agent properties, inter-agent interactions, the influence of the environment on the agents' actions, and ongoing evolutionary processes on a part of the agents as well as the environment.
Chan[57], Chan et al.[58], Lakhal et al.[59]	Agent-Based Simulation (ABS) is to simulate interactions of autonomous objects (called agents) to identify, explain, generate, and design emergent behaviors.
Singh et al.[60]	"We simulate two types i.e., type IIa and type IIb of emergent behavior as these are the most interesting with respect to engineering applications." Note: Type IIa and IIb are subcategories of weak emergence.[18]
Yang et al.[61]	"Many emergent behaviors burst out in different simulations accompanying the changing parameters and different scenarios."
Tretmans [62]	The goal of model-based testing is to reduce the integration and test effort of industrial systems.

Pourafzal and Fereidunian[63] use three differential equations known as Lorenz equations in which adopting different parameters dictate the phase transition between order and chaos, meaning the transition between complex and chaotic systems as well as strong and spooky emergence.

Shin et al.[64] use a fault database to incrementally build a framework to increase the detection of emergence, being an iterative process to increase a common understanding.

Discussion

The endeavor of studying and understanding the phenomenon of emergence has been researched throughout centuries and will keep being investigated for centuries to come.[3] The Systems Engineering process and methods are a good start for detecting emergence by incorporating work and tests making this possible.[40] Raz and Raman[40-45] state that modeling can help us regarding simple emergence, while simulation is required to detect weak emergence.[12,40-45] We can update our modeling efforts when we understand the simulated emergent behavior, to evolve the system design and exploit and/or mitigate the detected emergence.

Several methods can be used within the Systems Engineering processes to detect emergent behavior. How the testing is set up, Design of Experiments (DoE), is crucial to facilitate detection of emergence by providing sufficient stimuli from component interactions, human input, and environmental impact to trigger the emergent behaviors of the system.[35] A system model with parameters related to the inputs and features of dependencies can help see the effect of changes and causes of emergence.[1] Modeling & simulation can detect emergence but depends on the correctness of the model.[55] Building a fault knowledgebase can help detecting emergence but will require building experience through an iterative learning process.[64] Statistical debugging is a software (SW) method that can help in detecting emergence.[46] Machine Learning (ML) is a method that can detect and predict potential future detection of emergence, but can also introduce emergence itself.[40]

The outcome of these findings for our ongoing research in detecting emergent behavior in engineered complex systems is that we should take advantage of existing methods like orthogonal arrays (fractional factorial design of experiments and Taguchi matrices) to systematically test what is relevant, as one of the paradigms to testing. Further, we should use modeling & simulation to provide system data. Finally, we need appropriate data analysis techniques (system model inspection, fault knowledgebase, statistical debugging, and machine learning) to extract relevant information from our system models and test data.

We want to identify and take advantage of the positive emergence, while detecting and mitigating the negative emergence. Many methods have shown promising results in different case studies.[1,4,13,15,19,35-61] The approaches range from intuitive specialized model views used by Guariniello et al.[1] to advanced non-intuitive methods like machine learning used by Raz et al.[40]. Different approaches will serve different cases to a varying degree, requiring a thorough evaluation process to select the most appropriate method to utilize in each case. In the following, we will discuss the proposed research questions, ref. section 1.3.

RQ2: *How does the literature explain and categorize emergence?*

The phenomenon of emergence has many descriptions, but they all roughly point in the same direction.[15,18-20,22,23] The main idea is that the macro level exposes behavior that cannot be traced back to the micro level. Different sources categorize emergence into defined forms and types, although they dispute the applicability of these and the impact of the observer. There are three main directions found in existing literature on emergence, being 1) unpredictable,[24] 2) predictable,[14,25] and 3) both.[12,18,20,26,27] The unpredictable direction covers strong and spooky types of emergence,[12] while the predictable direction covers simple and weak types of emergence.[12]

For our ongoing research, we describe emergence as behavior at macro level that cannot be traced down to micro level. We categorize emergence as predictable (simple and weak) as well as unpredictable (strong and spooky), and we look at it subjectively from an observer point of view.

RQ3: *What approaches and methods do we see in the literature for detecting emergence?*

There are several methods within three main approaches for detecting emergence. The first approach is to focus on what we test to establish a reasonable test coverage.[35] Orthogonal arrays is a method used to systematically test to a given interaction level,[36-38] and using temporal logic is another related method.[39] The second approach is related to how efficient we are able to execute a test cycle, looking into the manual vs automatic task balance.[4,13,19,47,48] Machine learning is a method of interest to automate data analysis,[40-45] and statistical debugging is another.[46] The third approach is modeling and simulation to increase our level of understanding through simulation and updated models.[15,49,51,54,61] MBSE is an overarching method being used,[52] as well as MBSI.[53] More specific methods used in modeling are system behavior modeling,[50] system interface modeling,[1] and conceptual modeling.[55] More specific methods used in simulation are model-based testing[62] and agent based simulation.[56-60]

Mogul[16] claims we will never be able to solve all emergent behavior of systems. This claim is reasonable, especially when we are dealing with complex systems and strong emergence, leading us into a potential infinite endeavor in pursuing all emergent behavior.

There is different cost effectiveness related to simulating for detection of different types of emergent behavior. Simple emergence can be managed through the system model,[12] and does not need to be simulated. Strong and spooky emergence are unpredictable and do not usually repeat,[12] meaning that the simulation effort may not be worthwhile. Weak emergence is then the type of emergence that can be found most effectively through simulation. Modeling & simulation cannot detect all emergent behaviors in the first iteration cycle, as we do not yet possess all necessary information,[55] We need additional iteration cycles to uncover more and more of the remaining emergent behaviors inherent in the system not revealed through previous iterations.[55]

For our continuing research we should focus on how to set up the required test suite to trigger the emergent behaviors of the system. In this process we must evaluate the need for different types of testing like virtual-, real-world-, hardware-, software-, and stress testing. This is a natural first step, being a prerequisite for any value in later data analysis. Further, we should focus on data from modeling & simulation iterations to increase the likelihood of detecting as much as possible of the emergence inherent in the system under test. We need to keep in mind that the capabilities of both the test arenas being used and the observers interpreting the results are impacting the subjectivity in the emergence categorization.

RQ1: *Where is the research in the area of detecting emergence in engineered complex systems?*

The current research coverage in the area of emergence stops in the transition between weak and strong type of emergence, weak emergence being predictable while strong emergence being unpredictable.[1,4,13,15,19,35-61] No strategies or methods have proven to be successful to detect strong emergence.[12] The literature is looking at simple and weak emergence in complicated and complex systems.[1,2,41-46,55,64] The perceived emergence and complexity are dependent on the observer.[12,25,26] The observer may see the emergence as strong at first, not understanding the emerging system behavior. After obtaining more knowledge through test and analysis, the observer may perceive the emergence as weak, understanding the emerging system behavior. The same observer impact applies for complexity categories, as in the transition from a perceived complex system to a complicated system.[25]

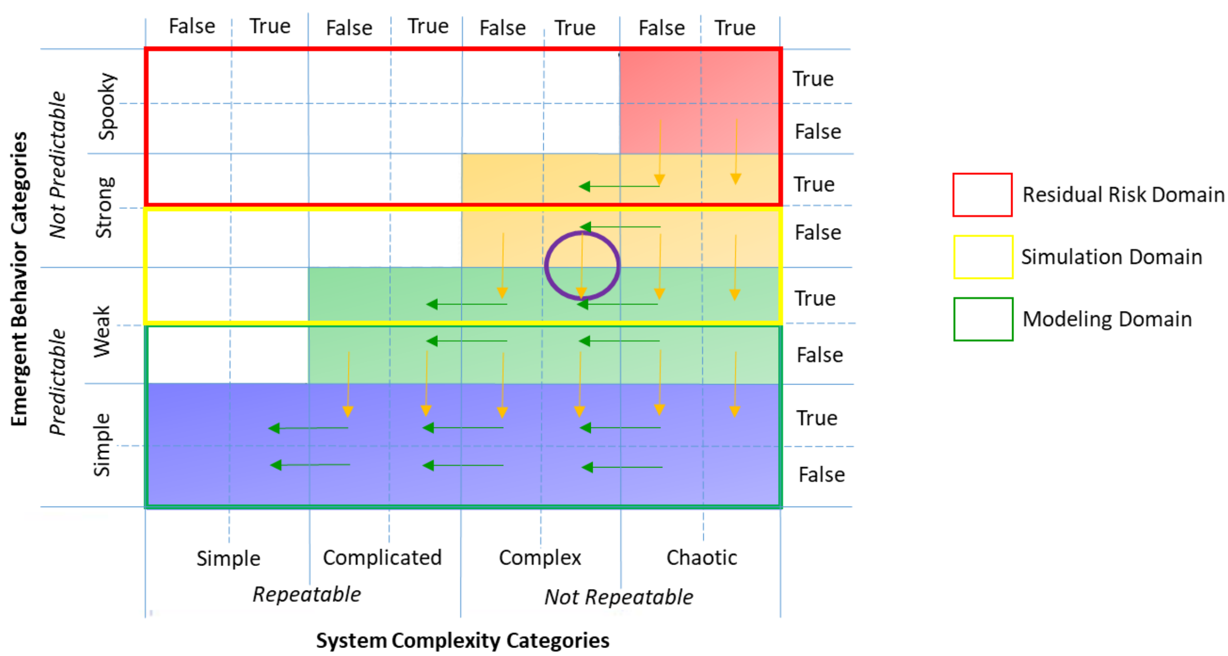

Figure 5: Relations and dynamics in emergence and complexity categorization

In this paper, we postulate that it is possible to build on existing practices and methods to detect false strong emergence in engineered complex systems. This is the part of strong emergence that has to do with the observer's perception due to lack of system knowledge. This will again help us to further develop best practices for emergence detection. By this false and true strong emergence separation, we are able to push the boundary of emergence detection to also include the part of perceived strong emergence that really is weak emergence. The transitions in the yellow region in *Figure 5* shows the part of emergence that could benefit both academia and industry to do more evolutionary research.

The direction for our continuing research will lay in the transition between predictable (weak) emergence and unpredictable (strong) emergence for complex engineered systems, see the purple circle in *Figure 5*. We will look into how the Systems Engineering process and methods can help the observer to increase knowledge sufficient to make the transition from perceived strong emergence to what is actually weak emergence. The aim is to create guidelines to facilitate the necessary increase in knowledge through simulation, including how to set up the test suite and how to perform the appropriate data analysis.

Conclusion

The industry developing engineered complex systems typically has problems detecting emergent system behavior, which are inherent in those kind of systems. This type of industry has a need to find better practices to detect more of this emergence.

The literature describes emergence as behavior emerging at macro level based on interactions on micro level that cannot be traced back to the micro level, but can also be triggered by lack of interactions at micro level.[15,18-20,22,23]

The literature further categorizes emergence in two forms (predictable and unpredictable) and four types (simple, weak, strong, and spooky) requiring different methods to detect.[12] Emergence can be either expected or unexpected, and beneficial or detrimental.[28] We typically want to detect the inherent emergent behavior of the system under test to be able to utilize the beneficial emergence and mitigate the detrimental emergence.

The literature shows several methods within three main approaches to detect emergence. First, test planning to evaluate test coverage. What to test is essential to trigger the inherent emergent behaviors of the system under test.[35] Orthogonal arrays[36-38] and temporal logic[39] are methods used. Second, manual vs automatic task balance evaluation for sufficient efficiency.[4,13,19,40-48] Machine learning is a method used for automatic monitoring of system effects and performance that can reveal emergent behavior so that mitigations can be made.[41-45] Third, modeling & simulation[15,49,51,54,61] iterations are necessary to capture and understand more and more of the weak emergence in the system.[55] Strong emergence can be detected and handled in the same way as weak emergence as long as it only has to do with the observer's lack of knowledge.[12] MBSE[52] and MBSI[53] are principle methodologies used. More specific methods used in modeling are system behavior modeling,[50] system interface modeling,[1] and conceptual modeling.[55] More specific methods used in simulation are model-based testing[62] and agent based simulation.[56-60]

The current research coverage in the area of emergence stops in the transition between weak and strong type of emergence.[1,4,13,15,19,35-61] The literature is looking at simple and weak emergence in complicated and complex systems.[1,2,41-46,55,64] True strong emergence and spooky emergence are difficult, if not impossible to detect in a repeatable way with the knowledge we possess today.[12] Continuous evolutionary research on detecting emergence should start at the transition between weak and strong emergence, building on proven strategies and methods for detecting weak emergence.

This literature study is limited in the sense of being a narrative synthesis and not a systematic review, which may result in not including all relevant literature on emergence.

Further research is needed to evaluate what strategies and methods that work to what degree to detect false strong emergence in engineered complex systems, transitioning to true weak emergence. Based on this output, we will be able to create best practices for that purpose.

Acknowledgements. We would like to acknowledge the Human Systems Engineering Innovation Framework (H-SEIF2) and the Self-Monitoring, Analysis, and Reporting Technology (SMART) Research Groups at the University of South-Eastern Norway for their contributions, providing valuable feedback to our proposed work during the survey period. In addition, we would like to acknowledge the Research Council of Norway for their funding. Finally, we would like to acknowledge the KONGSBERG company for sharing relevant industry information through their subject matter experts and databases.

References

1. Guariniello C, Khalid RA, Fang Z, Delaurentis D. System-of-systems tools and techniques for the analysis of cyber-physical systems. *Systems Engineering*. 2020;23(4):480-491. doi:10.1002/sys.21539
2. Kjeldaas KA, Haugen RA, Syverud E. Challenges in Detecting Emergent Behavior in System Testing. Wiley; 2021:1211-1228.
3. Ross WD, & Aristotle. *Aristotle's Metaphysics: A revised text with introduction and commentary*. Clarendon Press; 1924.
4. Haugen RA, Mansouri M. Applying Systems Thinking to Frame and Explore a Test System for Product Verification; a Case Study in Large Defence Projects. Wiley; 2020:78-93.
5. Honour E. Systems Engineering Return on Investment. Defense and Systems Institute, University of South Wales. Accessed May 23, 2022. http://www.hcode.com/seroi/index.html
6. Sutton A, Clowes M, Preston L, Booth A. Meeting the review family: exploring review types and associated information retrieval requirements. *Health Info Libr J*. Sep 2019;36(3):202-222. doi:10.1111/hir.12276
7. Randolph J. A Guide to Writing the Dissertation Literature Review. Practical Assessment, Research, and Evaluation2009.
8. Clarivate. Web of Science. Accessed 8th November, 2021. https://clarivate.com/webofsciencegroup/solutions/web-of-science/
9. Elsevier. Scopus. Accessed 9th November, 2021. https://www.scopus.com/home.uri

10. IEEE. IEEE Explore. Accessed 10th November, 2021. https://ieeexplore.ieee.org/Xplore/home.jsp
11. Page MJ, McKenzie JE, Bossuyt PM, et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*. Mar 29 2021;372:n71. doi:10.1136/bmj.n71
12. Mittal S, Diallo S, Tolk A. *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*. Wiley; 2018:1-395.
13. Szabo C, Teo Y. An integrated approach for the validation of emergence in component-based simulation models. 2012:2739-2749.
14. Bedau MA. Weak emergence. *Nous*. 1997;31(s11):375-399. doi:10.1111/0029-4624.31.s11.17
15. Holland J. *Emergence, From Chaos to Order*. Basic Books; 1999.
16. Mogul JC. Emergent (mis)behavior vs. complex software systems. Association for Computing MachineryNew YorkNYUnited States; 2006:293-304.
17. Callister PS, Andersson J. Evaluation of System Integration and Qualification Strategies using the Technical Debt metaphor; a case study in Subsea System Development. Wiley; 2015:
18. Fromm J. Types and forms of emergence. *arXiv: Adaptation and Self-Organizing Systems*. 2005;
19. Giammarco K. Practical modeling concepts for engineering emergence in systems of systems. 2017:1-6.
20. Holland OT. Taxonomy for the modeling and simulation of emergent behavior systems. In: *Agent Directed Simulation Symposium, ADS 2007 - Proceedings of the 2007 Spring Simulation Multiconference, SpringSim 2007*. Association for Computing Machinery, Inc; 2007:28-35.
21. Kadri H, Lakhal O, Conrard B, Merzouki R. Formal approach to SoS management design. 2021:
22. Kopetz H. Emergence. *Real-Time Systems*. 2 ed. Springer; 2011:43-45:chap 2.4. *Real-Time Systems Series*.
23. Wolf T, Holvoet T. Emergence versus self-organization: different concepts but promising when combined. Springer; 2004:1-15.
24. McLaughlin B. The Rise and Fall of British Emergentism. In: Beckermann A, Flohr H, Kim J, eds. *Emergence or Reduction?: Essays on the Prospects of Nonreductive Physicalism*. De Gruyter; 1992:49-93.
25. Mnif M, Muller-Schloer C. Quantitative Emergence. 2006:78-84.
26. Mittal S, Rainey L. Harnessing emergence: the control and design and emergent behavior in system of systems engineering. 2015:
27. Seth AK. Measuring emergence via nonlinear granger causality. 2008:545-553.
28. Kopetz H, Bondavalli A, Brancati F, Frömel B, Höftberger O, Iacob S. Emergence in Cyber-Physical Systems-of-Systems (CPSoSs). In: Bondavalli A, Bouchenak S, Kopetz H, eds. *Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy*. Springer International Publishing; 2016:73-96.
29. Johnson CW. What are emergent properties and how do they affect the engineering of complex systems? *Reliability Engineering and System Safety*. 2006;9(12):1475-1481. doi:10.1016/j.ress.2006.01.008

30. Zeigler BP. A note on promoting positive emergence and managing negative emergence in systems of systems. *The Journal of Defense Modeling and Simulation*. 2016;13(1):133-136. doi:10.1177/1548512915620580
31. Adcock R, Jackson S, Fairley D, Singer J, Hybertson D. Cloutier RJ, ed. *"Emergence" in The Guide to the Systems Engineering Body of Knowledge (SEBoK)*. 2021. Accessed 31st October. http://www.sebokwiki.org
32. Freund L, Al-Majeed S, Millard A. Towards the Definition of a Strategic Complexity Management Framework for Complex Industrial Systems. 2021:
33. Freund L, Al-Majeed S, Millard A. Case Studies Key-Findings of a Strategic Complexity Management Framework for Industrial Manufacturing Systems. 2021:
34. Snowden D. Cognitive Edge. Accessed 25th November, 2021. https://www.cognitive-edge.com/
35. Haugen RA, Ghaderi A. Modelling and Simulation of Detection Rates of Emergent Behaviors in System Integration Test Regimes. Vitenskapelig artikkel. *Linköping Electronic Conference Proceedings*. 2021;
36. Dunn K. *Process Improvements Using Data*. 2021.
37. Montgomery DC. *Design and Analysis of Experiments, 8th ed.* Wiley; 2017.
38. Taguchi G, Jugulum R, Taguchi S. *Computer-based robust engineering : essentials for DFSS*. ASQ Quality Press; 2004.
39. Khan TA, Jing W. On formalization of emergent behaviors in multiagent systems with limited interactions. 2016:0553-0558.
40. Raz AK, Llinas J, Mittu R, Lawless W. Engineering for Emergence in Information Fusion Systems: A Review of Some Challenges. In: *FUSION 2019 - 22nd International Conference on Information Fusion*. Institute of Electrical and Electronics Engineers Inc.; 2019:
41. Raman R, Jeppu Y. An Approach for Formal Verification of Machine Learning based Complex Systems. 2019:544-559.
42. Raman R, Jeppu Y. Formal validation of emergent behavior in a machine learning based collision avoidance system. In: *SYSCON 2020 - 14th Annual IEEE International Systems Conference, Proceedings*. Institute of Electrical and Electronics Engineers Inc.; 2020:
43. Raman R, Gupta N, Jeppu Y. Framework for Formal Verification of Machine Learning Based Complex System-of-System. Wiley; 2021:
44. Raman R, Jeppu Y. Does the Complex SoS Have Negative Emergent Behavior? Looking for Violations Formally. In: *15th Annual IEEE International Systems Conference, SysCon 2021 - Proceedings*. Institute of Electrical and Electronics Engineers Inc.; 2021:
45. Murugesan A, Raman R. Reinforcement Learning for Emergent Behavior Evolution in Complex System-of-Systems. IARIA; 2021:
46. Diallo SY, Lynch CJ, Gore R, Padilla JJ. Emergent behavior identification within an agent-based model of the Ballistic Missile Defense System using statistical debugging. Article. *J Def Model Simul-Appl Methodol Technol-JDMS*. Jul 2015;13(3):275-289. doi:10.1177/1548512915621973
47. Øvergaard A, Muller G. System Verification by Automatic Testing. Wiley; 2013:356-367.
48. Enoiu E, Sundmark D, Causevic A, Pettersson P. A Comparative Study of Manual and Automated Testing for Industrial Control Software. ResearchGate; 2017:412-417.

49. Bondar S, Hsu JC, Pfouga A, Stjepandić J. Agile digital transformation of System-of-Systems architecture models using Zachman framework. Article. *J Ind Inf Integr*. Sep 2017;7:33-43. doi:10.1016/j.jii.2017.03.001
50. Kossiakoff A, Flanigan DA, Seymour SJ, Biemer MS. *Systems Engineering Principles and Practice*. 3rd ed. Systems Engineering & Management. Wiley; 2020.
51. Neace KS, Chipkevich MBA. Designed Complex Adaptive Systems Exhibiting Weak Emergence. In: *Proceedings of the IEEE National Aerospace Electronics Conference, NAECON*. Institute of Electrical and Electronics Engineers Inc.; 2018:214-221.
52. Zeigler B, Mittal S, Traore M. MBSE with/out Simulation: State of the Art and Way Forward. Article. *Systems*. Dec 2018;6(4):18. 40. doi:10.3390/systems6040040
53. Montgomery PR. Model-Based System Integration (MBSI) - Key Attributes of MBSE from the System Integrator's Perspective. 2013:
54. Cummings MA. *Identifying and quantifying emergent behavior through system of systems modeling and simulation*. 2015. Accessed 17th February. https://calhoun.nps.edu/handle/10945/47243
55. Paunovski O, Eleftherakis G, Cowling AJ. Disciplined exploration of emergence using multi-agent simulation framework. Article. *Comput Inf*. 2009;28(3):369-391.
56. Kubic A. Towards a Formalization of Emergence. *Journal of Artificial Life*. 2003;9(1):41-65. doi:10.1162/106454603321489518
57. Chan WKV. Interaction metric of emergent behaviors in agent-based simulation. 2011:357-368.
58. Chan WKV, Son Y, Macal CM. Agent-based simulation tutorial - simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. 2010:135-150.
59. Lakhal O, Koubeissi A, Aitouche A, Sueur C, Merzouki R. Autonomous navigation through a system of systems cooperation. 2021:
60. Singh S, Lu S, Kokar MM, Kogut PA, Martin L. Detection and classification of emergent behaviors using multi-agent simulation framework (WIP). Society for Modeling and Simulation International (SCS); 2017:
61. Yang KW, Chen YW, Lu YJ, Zhao QS. The study of guided emergent behavior in system of systems requirement analysis. In: *2010 5th International Conference on System of Systems Engineering, SoSE 2010*. 2010:
62. Tretmans J. *Tangram: Model-based integration and testing of complex high tech systems*. 2007.
63. Pourafzal A, Fereidunian A. A Complex Systems Approach To Feature Extraction For Chaotic Behavior Recognition. 2020:1-6.
64. Shin S, Hyun S, Shin Y, Song J, Bae D. Uncertainty based Fault Type Identification for Fault Knowledge Base Generation in System of Systems. 2021:
65. Unit. ORIA. Accessed 12th November, 2021. https://bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/search?vid=HIT

Biography

Rune André Haugen is an industrial-PhD candidate at the University of South-Eastern Norway (USN). He was in service with the Royal Norwegian Air Force (RNoAF) from 1997 to 2003, including graduation from the RNoAF Officer Candidate School in Stavern (1999) and the RNoAF Academy in Trondheim (2001). He holds both a BSc (2006) and a MSc (2013) in Systems Engineering from USN. He has worked as a design engineer at FMC Kongsberg Subsea from 2006 to 2008 (3D modeling), and as a system engineer at Kongsberg Defence and Aerospace since 2008 (system design and system test).

Elisabet Syverud is an Associate Professor in systems engineering at the University of South-Eastern Norway. She has 20 years of work experience from the industry, including three years at KDAs missile division as part of the Systems Engineering group. She received her MSc in Aerospace Engineering from the University of Kansas, US, and her PhD in Thermal Energy from the Norwegian University of Science and Technology in Trondheim, Norway. She started her industrial career in 1993 and has worked in multiple roles in the oil & gas and defense industries for almost 20 years. Since 2019, she is Associate Professor of Systems Engineering and Head of Department Science and Industry Systems at USN.

Gerrit Jan Muller, originally from the Netherlands, received his MSc in physics from the University of Amsterdam in 1979. He worked from 1980 until 1997 at Philips Medical Systems as a system architect, followed by two years at ASML as a manager of systems engineering, returning to Philips (Research) in 1999. Since 2003 he has worked as a senior research fellow at the Embedded Systems Institute in Eindhoven, focusing on developing system architecture methods and the education of new system architects, receiving his PhD in 2004. In January 2008, he became a full professor of systems engineering at the University of South-Eastern Norway. He continues to work as a senior research fellow at the Embedded Systems Institute in Eindhoven in a part-time position. All information (System Architecture articles, course material, curriculum vitae) can be found at: Gaudí systems architecting http://www.gaudisite.nl/

Nils-Olav Skeie got his MSc in Cybernetics from Norwegian University of Science and Technology (NTNU) in 1985. He worked with system development within the computer, aviation and maritime industry for more than 20 years before receiving a PhD within machine learning in 2008 in a cooperation between NTNU and the University of South-Eastern Norway (USN). In 2006 he went back to the academia and has been teaching BSc, MSc and PhD students in software engineering and system engineering. He continued to work as a part time system architect for the maritime industry from 2008 to 2015. He became a professor of industrial machine learning at USN in 2020.

Appendix A: Literature Search Results

Table 6: Primary references found through database searches

Index	Primary references	Citations	Publication channel	Affiliations
1	Bondar, Hsu, Pfouga and Stjepandić [49]	26	Journal of Industrial Information Integration (level 1)	PROSTEP California State University
2	Chan, Son and Macal [58]	53	Proceedings of the 2010 Winter Simulation Conference (level 1)	Rensselaer Polytechnic Institute The University of Arizona Argonne National Laboratory
3	Chan [57]	17	Proceedings of the 2011 Winter Simulation Conference (WSC) (level 1)	Rensselaer Polytechnic Institute
4	Diallo, Lynch, Gore and Padilla [46]	5	Journal of Defense Modeling and Simulation-Applications Methodology Technology-Jdms (level 1)	Old Dominion University VMASC of ODU
5	Holland [20]	27	Agent Directed Simulation Symposium, ADS 2007 (level 0)	Naval Surface Warfare Center, Dahlgren Division

Index	Primary references	Citations	Publication channel	Affiliations
6	Khan and Jing [39]	1	2016 IEEE International Conference on Electro Information Technology (EIT) (level 1)	Bradley University
7	Mittal, Diallo and Tolk [12]	8	Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach (level 2)	MITRE Corporation Old Dominion University
8	Mnif and Muller-Schloer [25]	30	2006 IEEE Mountain Workshop on Adaptive and Learning Systems (level 0)	University of Hannover
9	Neace and Chipkevich [51]	0	IEEE National Aerospace and Electronics Conference, NAECON 2018 (level 1)	Johns Hopkins University
10	Paunovski, Eleftherakis and Cowling [55]	9	Computing and Informatics (level 1)	South-East European Research Centre (SEERC) City College University of Sheffield
11	Pourafzal and Fereidunian [63]	0	2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS) (level 0)	Toosi University of Technology
12	Raman and Jeppu [42]	2	14th Annual IEEE International Systems Conference, SYSCON 2020 (level 1)	Honeywell Technology Solutions Lab
13	Raman and Jeppu [44]	0	15th Annual IEEE International Systems Conference, SysCon 2021 (level 1)	Honeywell Technology Solutions Lab
14	Raz, Llinas, Mittu and Lawless [40]	1	22nd International Conference on Information Fusion, FUSION 2019 (level 1)	Purdue University University at Buffalo United States Naval Research Laboratory Paine College
15	Yang, Chen, Lu and Zhao [61]	11	5th International Conference on System of	National University of Defense Technology

Index	Primary references	Citations	Publication channel	Affiliations
			Systems Engineering, SoSE 2010 (level 1)	
16	Zeigler, Mittal and Traore [52]	12	Systems (level 1)	University of Arizona MITRE Corporation University of Bordeaux

Table 7: Secondary references found through references in the primary references

Index	Secondary references	Citations	Publication channel	Affiliations
1	Bedau [14]	97	Nous journal (level 2)	Reed College
2	Fromm [18]	66	arXiv (level 0)	University of Kassel
3	Giammarco [19]	3	12th International Conference on System of Systems Engineering, SoSE 2017 (level 1)	Naval Postgraduate School
4	Holland [15]	1053	Basic Books (level 2)	University of California
5	Johnson [29]	160	Reliability Engineering and System Safety journal (level 2)	University of Glasgow
6	Kubic [56]	117	Journal of Artificial Life (level -)	Silesian University
7	Mittal and Rainey [26]	51	SummerSim: summer simulation multi-conference (level 0)	Dunip Technologies Integrity Systems and Solutions
8	Mogul [16]	140	Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems (level 0)	HP Laboratories Palo Alto
9	Murugesan and Raman [45]	0	ICONS: The Sixteenth International Conference on Systems 2021 (level 1)	Honeywell Aerospace Honeywell Technology Solutions Lab
10	Raman and Jeppu [41]	5	14th Annual IEEE International Systems Conference, SYSCON 2020 (level 1)	Honeywell Technology Solutions Lab
11	Raman, Gupta and Jeppu [43]	0	INCOSE International Symposium 2021 (level 1)	Honeywell Technology Solutions Lab
12	Seth [27]	30	Artificial Life XI: Proceedings of the	University of Sussex

Index	Secondary references	Citations	Publication channel	Affiliations
			Eleventh International Conference on the Simulation and Synthesis of Living Systems 2008 (level 2)	
13	Szabo and Teo [13]	14	Proceedings of the 2012 Winter Simulation Conference (level 1)	University of Adelaide National University of Singapore
14	Wolf and Holvoet [23]	360	International Workshop on Engineering Self-Organizing Applications 2004 (level 0)	University of Leuven
15	Zeigler [30]	11	The Journal of Defense Modeling and Simulation (level 1)	University of Arizona MITRE Corporation

Table 8: Tertiary references found through other means of information

Index	Tertiary references	Citations	Publication channel	Affiliations
1	Adcock, Jackson, Fairley, Singer and Hybertson [31]	-	Systems Engineering Body of Knowledge (level 2)	Stevens Institute of Technology
2	Callister and Andersson [17]	2	INCOSE International Symposium 2016 (level 1)	Aker Solutions University of South-Eastern Norway
3	Cummings [54]	3	PhD dissertation (level -)	Naval Postgraduate School
4	Enoiu, Sundmark, Causevic and Pettersson [48]	6	IEEE International Conference on Software Testing, Verification and Validation, ICST 2017 (level 1)	Maelardalen University
5	Freund, Al-Majeed and Millard [32]	0	16th International Conference on System of Systems Engineering, SoSE 2021 (level 1)	University of Lincoln
6	Freund, Al-Majeed and Millard [33]	0	16th International Conference on System of Systems Engineering, SoSE 2021 (level 1)	University of Lincoln

Index	Tertiary references	Citations	Publication channel	Affiliations
7	Guariniello, Khalid, Fang and Delaurentis [1]	4	Systems Engineering journal 2020 (level 2)	Purdue University University of Science and Technology, Wuhan, China
8	Haugen and Ghaderi [35]	0	SIMS EUROSIM 2021 (level 1)	University of South-Eastern Norway
9	Haugen and Mansouri [4]	2	INCOSE International Symposium 2020 (level 1)	University of South-Eastern Norway Stevens Institute of Technology
10	Kadri, Lakhal, Conrard and Merzouki [21]	0	16th International Conference on System of Systems Engineering, SoSE 2021 (level 1)	Lille University
11	Kjeldaas, Haugen and Syverud [2]	1	INCOSE International Symposium 2021 (level 1)	University of South-Eastern Norway
12	Kopetz [22]	2139	Real-Time Systems 2011 (level 2)	Vienna University of Technology
13	Kopetz, Bondavalli, Brancati, Frömel, Höftberger and Iacob [28]	7	Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy 2016 (level 2)	Vienna University of Technology University of Florence Resiltech SRL Thales Delft
14	Kossiakoff, Flanigan, Seymour and Biemer [50]	-	Systems Engineering Principles and Practice (level 2)	Johns Hopkins University
15	Lakhal, Koubeissi, Aitouche, Sueur and Merzouki [59]	0	16th International Conference on System of Systems Engineering, SoSE 2021 (level 1)	University of Lille Junia Institute Centrale Lille Institute
16	McLaughlin [24]	358	Emergence or Reduction?: Essays on the Prospects of Nonreductive Physicalism 1992 (level 2)	Rutgers
17	Montgomery [53]	8	Conference on Systems Engineering Research, CSER 2013 (level 1)	Naval Postgraduate School

Index	Tertiary references	Citations	Publication channel	Affiliations
18	Shin, Hyun, Shin, Song and Bae [64]	0	16th International Conference on System of Systems Engineering, SoSE 2021 (level 1)	Korea Advanced Institute of Science and Technology (KAIST)
19	Singh, Lu, Kokar, Kogut and Martin [60]	8	Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2017 (level 0)	North-eastern University Lockheed Martin
20	Snowden [34]	-	The Cynefin Co (level -)	Cognitive Edge
21	Øvergaard and Muller [47]	0	INCOSE International Symposium 2013 (level 1)	University of South-Eastern Norway
22	Dunn [36]	2	Process Improvements Using Data (level -)	DH Pace Company, Inc
23	Montgomery [37]	2	Design and Analysis of Experiments, 8th ed. (level -)	Arizona State University
24	Taguchi, Jugulum and Taguchi [38]	2	Computer-Based Robust Engineering : An Essential for DFSS (level -)	Aoyama Gakuin University Massachusetts Institute of Technology (MIT) ASI Consulting Group
25	Tretmans [62]	-	Tangram: Model-based integration and testing of complex high-tech systems (level -)	Embedded Systems Institute, Eindhoven

Note: Level 0 means unrecognized publishing channel probably not categorized to level 1 or 2. Level – means unrecognized publishing channel probably categorized to level 1 or 2.

Appendix B: Literature Search Log

Table 9: Search log for literature search in relevant databases

Search date	**Database** (for example Oria, ERIC)	**Search words combined with AND, OR, NOT** (for example AND combined search words and limits the results. OR is used between alternative search words and gives you more results)	**Limitations** (for exemple language, publication year, peer reviewed)	**Results** (after the search words are combined)	**Comments /Notes**
08.11.21	Web of Science (WoS)	TS=(Emergen* AND Behavio* AND Complex* AND System* AND Engineering AND Model* AND Simulati*)		51 15 (title)	-
08.11.21	Web of Science (WoS)	TS=(Emergen* AND Behavio* AND Complex* AND System* AND "Machine Learning")		41 8 (title)	-
08.11.21	Web of Science (WoS)	TS=(Emergen* AND Behavio* AND System* AND Engineering* AND Model* AND Simulati* AND (Identif* OR detect*)		19 3 (title)	2 duplicates (title)
08.11.21	Web of Science (WoS)	KP=(Emergen* AND behavio* AND (Identif* OR detect*))		36 0 (title)	-
08.11.21	Web of Science (WoS)	AK=(Emergen* AND behavio* AND (Identif* OR detect*))		8 4 (title)	-

Search date	**Database** (for example Oria, ERIC)	**Search words combined with AND, OR, NOT** (for example AND combined search words and limits the results. OR is used between alternative search words and gives you more results)	**Limitations** (for exemple language, publication year, peer reviewed)	**Results** (after the search words are combined)	**Comments /Notes**
	Web of Science			155 28 (title) 15 (abs) 5 (text)	1 roadmap
09.11.21	SCOPUS	TITLE-ABS-KEY (*emergen** AND *behavio?r** AND *complex** AND *system* AND engineering* AND *model** AND *simulat**)		50 13 (title)	-
09.11.21	SCOPUS	TITLE-ABS-KEY (*emergen** AND *behavio** AND *complex** AND *system** AND *"Machine Learning"*)		107 9 (title)	-
09.11.21	SCOPUS	*TITLE-ABS-KEY (*emergen* *AND* behav io* *AND* system* *AND* engineering *AND* model* *AND* simulat* *AND* identif* OR detect**)*		114 7 (title)	-
	SCOPUS			271 29 (title) 25 (abs) 8 (text)	8 duplicates with WoS 2 roadmap

Search date	**Database** (for example Oria, ERIC)	**Search words combined with AND, OR, NOT** (for example AND combined search words and limits the results. OR is used between alternative search words and gives you more results)	**Limitations** (for exemple language, publication year, peer reviewed)	**Results** (after the search words are combined)	**Comments /Notes**
10.11.21	IEEE	("All Metadata":Emergen*) AND ("All Metadata":Behavio*) AND ("All Metadata":Complex) AND ("All Metadata":System*) AND ("All Metadata":Engineering) AND ("All Metadata":Model*) AND ("All Metadata":Simulat*)		119 8 (title)	-
10.11.21	IEEE	("All Metadata":Emergen*) AND ("All Metadata":Behavio*) AND ("All Metadata":Complex) AND ("All Metadata":System*) AND ("All Metadata":"Machine Learning")		28 5 (title)	1 duplicate (title)
10.11.21	IEEE	("All Metadata":Emergen*) AND ("All Metadata":Behavio*) AND ("All Metadata":System*) AND ("All Metadata":Engineerin		72 6 (title)	2 duplicates (title)

Search date	**Database** (for example Oria, ERIC)	**Search words combined with AND, OR, NOT** (for example AND combined search words and limits the results. OR is used between alternative search words and gives you more results)	**Limitations** (for exemple language, publication year, peer reviewed)	**Results** (after the search words are combined)	**Comments /Notes**
		g) AND ("All Metadata":Model*) AND ("All Metadata":Simulati*) AND ("All Metadata":(Identifi* OR detect*))			
	IEEE			219 16 (title) 16 (abs) 5 (text)	5 duplicates with Scopus

Note: We could preferably update the search words to do this screening process more automatically by using the NOT operator. Some records were duplicates in between the different search words we used in the same database, and some records were duplicates in between the different databases. A few of the records were put on a roadmap for later study because they were focusing more on the tools than the methods. For some records, we were not able to retrieve the full text documents from the USN database.[65]