Evolvability of Representations in Complex System Engineering: a Survey

Andreas Richter
Bielefeld University
Postfach 100 131
D-33501 Bielefeld, Germany
anrichter@techfak.uni-bielefeld.de

Mario Botsch
Bielefeld University
Postfach 100 131
D-33501 Bielefeld, Germany
botsch@techfak.uni-bielefeld.de

Stefan Menzel
Honda Research Institute Europe GmbH
Carl-Legien-Str. 30
D-63073 Offenbach/Main, Germany
stefan.menzel@honda-ri.de

***Abstract*—A successful design optimization crucially depends on the underlying representation, which has to adapt to a variety of demands and changing boundary conditions. Complex system engineering addresses these challenges through key features like self-organization, modularity, locality, or evolution. The representation covers the parameter setup (*location* and *quantity*) and the mapping between parameter space (*genotype*) and design space (*phenotype*), and should allow for both adaptation and specialization of a design. To quantify the potential of a representation, suitable quality criteria are needed. Evolvability is such a criterion, which has been derived from biological analysis. However, many biological and technical studies propose different definitions of evolvability. We analyze, interpret, and extend them in order to derive an evolvability criterion suitable for complex system engineering. This can be used as a basis for future design optimization problems.**

I. Introduction

The increasing complexity of engineering systems, as for instance in automotive design scenarios, requires advanced optimization methods to efficiently provide high-quality solutions for real-world problems. The dynamic interaction of different physical domains (e.g., drag minimization, stiffness optimization, manufacturing constraints) as well as changing customer demands (e.g., shape features) have to be considered simultaneously and—in the optimal case—holistically.

Classic analysis focuses on specific, independent subproblems only, for instance optimization of representation parameters, modeling of quality criteria, or the choice of the optimization routine. In contrast, in complex system engineering relevant features are considered *simultaneously*, because they are strongly linked and they interact with each other. The changing environment requires highly flexible setups, which have to offer room for adaptation and should even accept unexpected behavior. A major task is the estimation and proper quantification of the adaptation potential of the employed representation. The motivation for this survey is to provide a quantitative measure for optimization potential and development capabilities of these digital prototypes. In our target application—automotive design optimization—a prototype is represented by a triangulated surface plus a so-called shape morphing operator, such as free-form deformation (FFD) or RBF morphing [1]. An example of both deformation methods is shown in Figure 1.

978-1-4799-7492-4/15/$31.00

In complex system engineering, the evaluation of different setups for the optimization of dynamically changing systems has been investigated under the biologically-inspired term *evolvability*. Evolvability, as a biological feature in evolutionary computation, characterizes the potential of an individual or a population in the evolutionary process. A population with high evolvability is expected to develop better in the current environment and to adapt faster to new conditions. In this context, evolvability takes into account the genotype-phenotype relationship as well as the influence of the (potentially changing) environment. But even in the field of evolutionary biology there has not been a unique definition of evolvability.

When transferring this concept to evolutionary optimization in engineering applications, evolvability can be used to measure the optimization potential of a representation. In our automotive design scenario the representation consists of the model parameters to be optimized (*genotype*), which induce a design variation (*phenotype*) through the shape deformation operator (*genotype-phenotype mapping*). By utilizing the idea of evolvability in complex system engineering, we want to analyze, quantify, and improve the representations, which eventually should lead to an increased performance of the resulting system. To this end we analyze different definitions of evolvability derived from biological analyses and experiments, examine studies based on these observations in technical engineering, and evaluate these approaches in the context of complex system engineering. The gained insight allows us to motivate a conceptual formulation of evolvability that is suitable as a quality criterion for representations in complex system engineering. To this end, our general concept

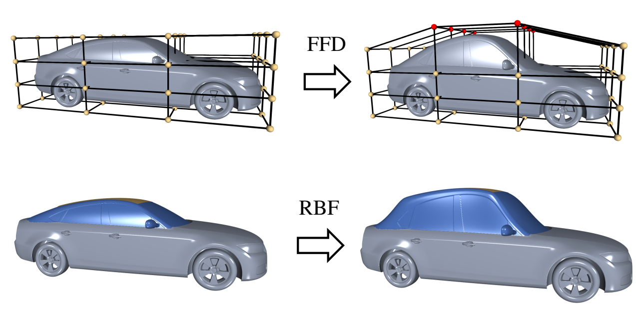

Fig. 1. A typical free-form deformation (top) and an RBF morph (bottom) in an automotive design scenario [1].

of evolvability has to be translated into application-specific criteria, e.g., in the context of automotive design optimization.

We start by discussing complex system engineering in Section II, where we describe key features, modeling approaches, and the importance of evolvability. Afterwards, we give an overview of different interpretations of evolvability in the biological context, and then bridge the gap from biological simulations to the optimization of technical problems (Section III). In Section IV we briefly discuss further aspects and topics related to evolvability. We summarize the gained experience in order to propose an evolvability concept for complex systems (Section V). A table that categorizes the evolvability-related articles covered in this paper finally gives a coarse overview of the literature.

II. Complex System Engineering

A system is a construct or collection of different elements that together produce results not obtainable by the elements alone [2]. Creating and linking these elements is the task of the engineer in order to fulfill the customer's needs throughout the entire life cycle of the system.

Complexity arises through many different objectives or conflicting interests, (nonlinear) interactions of components, large design spaces, feedback loops, or adaptive processes [3]. Varying fields of research are involved, such as engineering (dynamical systems and their control), computer science (modeling and simulation), biology (self-organized systems), or physics (physical models) [4]. Complex systems are based on simple, maybe different, but separated components. They are combined and linked together in order to achieve the multiple, unpredictable, and time-varying goals. The system should be able to change the representation or even the fundamental structure. Examples are communication or transportation networks, financial markets, organisms, or insect colonies [5].

In *classical* engineering the designer gathers information to specify the problem as precisely as possible. Uncertainty is eliminated as much as possible. As a consequence many pieces of information are needed, e.g., about the conditions of the environment and the task that should be performed. This requires specialized knowledge and competence of the designer, who has to model the functionality and the overall process. The typical approach is to simplify a system as much as possible. Only when the designer tested the specialized system well enough, it is completely fixed and reproduced. This leaves no space for later adaptation other than intentionally planned by the designer. The *classical* goal is to obtain a single specialized solution that can be reliably reproduced. The required predictability, transparency, and controllability inevitably prevents self-adaptation [5].

Complex systems, in contrast, have to operate in unknown, uncertain, unforeseeable, dynamic environments. The focus is set on the adaptation potential to handle these demands. The required flexibility is gained by simple, local, and linked processes, which in concert solve the global problem. The designer models these simple processes and their connection, and thereby produces a "blueprint" [4]. The system is responsible for the setup of the processes and their re-evaluation and adaptation during the life time of the system (self-organization and evolution). Thus the goal in complex system engineering is to develop a method that enables the system to autonomously interact with its environment; or as it is stated in Mina et al. [5]: "becoming" is "being".

For this approach two main characteristics are important: *self-organization* and *evolution* [5]. *Self-organization* is a large-scale and local organization of many simple components. The term large-scale describes the interaction of different conceived components with varying complexity. Like many other concepts, self-organization is inspired by nature, e.g., in molecules many atoms form a large structure through local forces. Although human-designed systems usually (and intentionally) ignore this feature, we unconsciously use self-organization in engineering problems: Small teams are built to solve sub-problems, new links between solutions are created, or problems are adjusted to new conditions. The result is that non-trivial, large-scale optimization can be produced by simple local processes [5], thereby leading to adaptive behavior without external command.

Engineering complex systems requires *evolutionary processes* to integrate self-organization [5]. The system is designed to solve unpredictable problems on its own with little information about the environment. The appropriate connection of components has to be re-adjusted or even the functionality of components has to extend over time to accomplish the varying goals. The evolutionary selection–mutation–recombination scheme handles these demands. The random mutation and recombination may vary the components and their linkage and the fittest configurations are selected with respect to the current problem (environment) by algorithms that implement these biological concepts. We want to state that evolutionary processes are not restricted to complex systems but advantageous in general engineering approaches.

The designer has to accept uncertainty as a system feature. It should be seen as a chance to generate a variety of unexpected new solutions. Depending on additional information he has the freedom to choose some of them. The dimension of the design space is seen as a benefit, since it increases the variation of design solutions. Hence any limitation of the design space has to be modeled carefully. A three-stage approach is proposed by Frei and Serugendo [3], where desired, allowed, and possible areas are specified. This induces a user-defined expectation, which cannot be set by the system itself. The engineer has to implement concepts that keep the system running within the specific area, but this at the same time restricts the development of the system. Thus a tradeoff has to be found, since the system should still be able to adapt its behavior.

Modularity as well as weak linkage are further concepts. Separated (modular) components can be modeled and exchanged easily if their dependencies are limited (weak linkage). The individual components/processes have to be fully functional even under changing conditions and have to be sufficiently flexible to achieve time-varying goals. This concept is referred to as *robust optimization* [6]; it can be used as one design methodology in complex system engineering. Another concept is *multi-functionality*, also known as degeneracy [7]: Multiple processes may perform the same task in one environment (redundancy), but work on different tasks under new conditions (flexibility).

Besides self-organization further self-*-properties may be added as characteristic properties of complex systems. Frei and Serugendo [3] propose self-(re)configuration (parameter adjustment over time) or self-repair (ability to correct failures). The major aspects of complex systems—modularity, simplicity and linkage of components, self-*-properties, or design space models—are examined in greater detail in [8, 9].

Given that particular research area, the important challenges for complex system engineering are:

- The simple components may be defined in a classical manner and their connections have to be set flexibly.
- The design space has to be modeled carefully for the (partly random) system evolution to find varying solutions.
- One has to focus on the optimal setup, rather than on the "optimal" solution.
- Convergence criteria are non-trivial to model, since the system's progress is unpredictable.

But how is evolvability involved in this process?

As the system has to reflect on itself, it requires meta-attributes to quantify the potential of its current configuration. These criteria are robustness, degeneracy, or adaptability, and they can be subsumed under the concept of evolvability. Evolvability improves or guarantees the progress of the system's development. It classifies the behavior of the system; it quantifies the design space that can be reached by the current representation; it guarantees the performance improvement of the system. In essence, evolvability covers the survivability, the solution variety, the adaptation potential, and the evolution speed in one single meta-attribute, and therefore is an important quality criterion.

With this comprehensive quality criterion we aim at measuring, optimizing, or adapting representation setups of complex systems based on customer demands, optimization targets, or environmental restrictions. But defining and measuring evolvability is a difficult challenge. In the next section we present a comprehensive analysis of biological and technical approaches in order to collect different aspects and modeling techniques for complex system engineering.

III. Evolvability

Evolution produces offspring of individuals through mutation and recombination. The new offspring should ideally be able to survive in the current environment and to adapt flexibly to environmental changes, since this improves the evolutionary development. In general, evolvability is meant to characterize the developmental potential or capability of individuals in the evolutionary process.

There is no unique precise definition of evolvability. Even in biology, where the term evolvability stems from, many different definitions have been proposed. Evolvability has been defined as the ability of a genotype to produce heritable phenotypic variation [10–12], the potential of a population for producing novel mutations for their use in the evolution of adaptations [13], or as a quantity to explain lineages of populations in the tree of life [14]. These are only some of the many different biological concepts. In general, evolvability describes the quality of biological evolution or the evolutionary capabilities of an individual or a population. It evaluates potential future benefits [10, 15]. Influencing factors are, e.g., the genotype-phenotype relation, the variety of phenotypes, or the speed of adaptation to natural (changing) environments.

Deriving from these approaches, and in agreement with Sterelny [16], we understand evolvability as a combination of the three attributes:

- *regularity*, which describes the quality of an individual or a population independent of the current environment,
- *variability*, which aims at a rich design space (phenotype),
- *adaptation potential*, which ensures the ability to adapt to changing conditions.

Note that we replaced the term *heritability* from Sterelny [16] by *regularity* in order to avoid conceptual conflicts, as heritability can itself be considered as a measure for the evolutionary potential (see the heritability vs. evolvability discussions in [17, 18]). In the following we discuss the evolvability concept by analyzing the individual attributes step by step.

A. Regularity

Formulating a suitable fitness function for the development of complex systems is a difficult and cumbersome process, since one cannot incorporate every individual quality aspect into the fitness criterion. This would inflate the fitness function, increase the computational cost of its evaluation, and would thereby slow down the overall optimization tremendously. One should therefore try to prevent the generation of infeasible (mortal) offspring *before* the environment (i.e., the optimization) evaluates it. We understand regularity as a stability attribute that reduces this infeasibility and thereby improves the evolutionary process. For instance, in automotive design a poor discretization of the car body will negatively influence any CFD simulation, therefore such configurations should be avoided. In the context of system engineering the restriction to feasible (regular) offspring is a limitation of the design space on the one hand, but on the other hand regularity can be interpreted as a safety guard.

The regularity of phenotypes is oftentimes not employed as an individual attribute, but rather incorporated into the attribute *robustness* (e.g. [19]), which we discuss later in Section IV-A. We explicitly emphasize and follow the approach of Sterelny [16], where regularity is declared as an extra trait, but also as a part of a more complex evolvability concept. The regularity criterion has to avoid problems that do not depend on the fitness function and which cannot be handled by the variability criterion. Sterelny [16] describes it as an *anti-outlaw condition*. A concrete example is given in [20], where control lattices for free-form deformation are constructed such that control points are well separated. This reduces the chance of flipping of control points and thereby avoids unfeasible self-intersections of the deformed object.

We cannot give a more precise definition of regularity, because it strongly depends on the representation and the actual optimization scenario. The designer has to define this attribute as an environmental-independent stability criterion. We incorporate it in our definition of evolvability in order to improve the fitness-independent performance of the system.

B. Variability

As we have shown before, complex systems operate in uncertain, dynamic environments. Therefore a criterion that measures and preserves the flexibility of the representation, independent of the environment, has to be incorporated in the definition of evolvability. This criterion should furthermore characterize the ability and potential to extensively explore the design space (phenotype). In biology many different synonyms exist for this concept, such as innovation, variation/variability, or even evolvability itself, as for example in [11, 12, 21]. Variability describes the future potential of obtaining varying phenotypes [11].

Wagner's approach to RNA analysis [12] reveals interesting properties and limitations of variability measures. This approach can be considered as a representative for a whole class of biological approaches that are based on the concept of *neutrality* [21–25]. Two genotypes are neutral if they map to the same phenotype, and they are neighboring when they are connected via a single point mutation (a mutation that changes just one parameter). The first variability definition of Wagner [12] is based on the neighborhood of a genotype (local definition), while the second definition analyzes the neighborhood of all neutral genotypes of a given phenotype (global definition). Both approaches compute the diversity of the phenotype and are purely discrete. The second one even requires global information of the phenotype. For complex systems this is a major drawback, since they have to operate with little information, and therefore cannot analyze the whole parameter space or design space. In the automotive scenario both spaces are continuous, which makes the definition of a neighborhood cumbersome and imprecise.

Jin and Trommler [26] solve this problem by replacing single point mutations by arbitrary mutations and by measuring the ratio of phenotype diversity to genotype variation. This requires a proper definition of distance metrics for both genotypes (parameter space) and phenotypes (design space). Lehmann and Menzel [20, 27] transfer this idea to a shape matching optimization using free-form deformation. Their variability criterion, defined as the ratio of phenotype variation to genotype variation, characterizes the quality of different representation setups and is used to improve the performance of the optimization. As the computation of this criterion requires global information, it is not really suitable for complex systems.

A possible solution can be derived from an approach called *novelty search*, which replaces the fitness selection criterion of evolutionary algorithms by a variability criterion. Example applications for maze navigation and biped robot experiments are described in [28–31]. Although we do not discuss particular algorithmic details in this survey, we briefly describe the idea for deriving variability: The variability of each individual of an offspring is measured by a novelty metric, which evaluates the dissimilarities to each other individual in the population resulting in a highly diverse population. To decide whether a concrete goal is reached one has to perform an additional objective-oriented evaluation. In a general (non-evolutionary) optimization context this approach can be seen as the generation and the evaluation of local samples for different representation setups.

The ability to extensively explore the design space is measured by the variability criterion, which characterizes the genotype-phenotype mapping. For complex system engineering we propose to define variability as the ratio of the potential phenotype diversity induced by genotype variation. It then is the designer's task to specify diversity metrics in both parameter space and design space as well as the mapping between them. Since global information is not available in complex systems, one has to fall back to local methods for computing variability.

C. Adaptation Potential

During the development of a complex system some parts may already be (close-to) optimal in the current state, while other parts have to adapt further to specific demands or changing conditions. For instance, in an automotive design process the roof may already be satisfactory, but the fender has to be improved with respect to drag. A representation that can only change both targets simultaneously is counterproductive. Enabling the representation to adapt to sensitive regions requires to incorporate a fitness-dependent criterion into the evolvability definition.

In the biological context several approaches identify evolvability itself with adaptation potential or adaptation speed of a population to an environment, e.g. [13, 32]. But this definition is rather imprecise, since a population is called adapted as soon as a beneficial trait occurs significantly more often. In the engineering context the optimization potential in a varying fitness landscape is investigated in [33–35]. In [36–38] the structural bias of an environment is analyzed. When different regions of the fitness landscape are linked, a representation that can learn that linkage will better adapt to changes in the environment than a representation that ignores the fitness landscape. This approach, however, requires knowledge about the different environments and the connections between them. We regard this as a contradiction to unpredictability in complex system engineering. The idea to include a fitness-dependent learning process is promising though. While it may slow down the development of the system in the beginning, it improves the long-term performance. Since fitness evaluations typically are computationally expensive, computationally cheaper surrogates can be used to approximate the original fitness function and replace it in the optimization [39]. While the surrogates are easier to evaluate by construction, they require an additional learning step.

Aulig [40] selects the representation that results in the best compromise for a variety of predefined environments and define this representation to have the highest adaptation potential. However, an additional evolutionary optimization for each environment has to be performed in order to evaluate each candidate and find the compromise, which can be rather costly. This approach can be considered as performing local optimizations for computing the adaptation potential.

Transferred to complex systems the adaptation potential characterizes the (fitness) improvement potential of a representation. One has to classify different representation setups either in a learning process, with an additional local optimization, or even with a combination of both. Based on available information about the environment this choice is left to the designer. In the spirit of complex system engineering we can also incorporate his/her experience, to specify important/sensitive regions for later adaptation.

D. Summary

Our literature survey leads to a concept for evolvability that is based on the three criteria regularity, variability, and adaptation potential, which—as we have discussed—are important in complex system engineering. How to exactly formulate them in a specific application context depends on the individual setting. Or, as Wagner stated in [12]: It is a matter of taste. Hence the designer has to analyze the system in order to formulate a particular quality attribute derived from these basic concepts. Based on the available information and resources the designer has to implement suitable methods for computing evolvability.

IV. Further Aspects of Evolvability

In the context of evolvability Many more attributes are used and can be investigated. In complex system engineering *robustness* is one important aspect, but its relation to evolvability is discussed contrarily. Another important aspect of complex systems is *modularity*. How it is integrated into our evolvability model is shown after discussing robustness. The setup of evolutionary algorithms influences the performance of the system as well. From that point of view evolvability can be defined even for algorithms, which we discuss in the third subsection.

A. Robustness vs. Evolvability

In optimization scenarios a solution is considered robust if noise does not affect its quality. The setup of an algorithm is considered robust if noise on the input data still leads to the same solution. Generally, the concept of robustness is important in complex systems to induce stability. In the biological context robustness reduces the mortality of offspring and therefore promotes the evolutionary process. Wagner and colleagues define it as the persistence of an organismal trait under perturbations [19]. The authors analyze robustness on three different levels: The first one is independent of the environment and is called stochastic noise. Robustness preserves the general quality if stochastic fluctuations occur in (biological) systems. The second level characterizes the influence of genetic variation on the phenotype. A trait is regarded as robust if genetic variation preserves it. The third level describes the survivability of a phenotype when changes of the environment occur. Our definition of evolvability can be considered to include robustness by interpreting stochastic noise as one part of regularity. Moreover, robustness to varying environments is covered by the attribute adaptation potential.

But whether robustness to genetic variation promotes or hinders variability is discussed contrarily. In biology robustness and evolvability are usually reduced to the variety of phenotypes [12, 21–25, 41–43]. Hence one could argue that a highly robust phenotype is not variable, but this statement is not always true. Wagner [12] defines robustness through the concept of neutrality in two ways: local and global (similar to the variability discussion in Section III-B). Robustness quantifies the neutrality in the neighborhood of a phenotype or the size of a whole set of neutral phenotypes, respectively. Interestingly, the local definitions of robustness and variability contradict each other, while the global ones agree: A large neutral set of phenotypes has many, typically diverse neighbors, and therefore also a higher variability.

The concept of cryptic gene variation [44] is a biological observation including both robustness and variability. When individuals are adapted to an environment their phenotype variation decreases, which reduces mortality because few non-adapted phenotypes occur. However, cryptic gene variation preserves the variation hidden in the genotypes, which promotes adaptation after a change in environmental conditions. In [32] a non-monotonic relation between robustness and variability is described, meaning that the most variable phenotype is medium robust. The authors argue that non-robust phenotypes cannot survive and very robust ones cannot evolve. In technical engineering this potential conflict is accepted, e.g., in [20, 27, 40, 45, 46]. The goal is to find variable solutions that are as robust as possible. The solutions on this Pareto front can be used according to the behavior of the environment. If it is stable, robust ones are preferred. If it is varying, variable ones are more promising.

The general conflict that robust phenotypes cannot adapt to new conditions is analyzed in [7, 47–49]. *Degeneracy* is proposed as the solution, and is defined as multi-functionality of components. For example, two different components (e.g., proteins) may perform the same task in the current environment, but different tasks once the environment changes. The switch between redundancy and diversity, which characterizes degeneracy, increases the robustness as well as the adaptation potential. If one component fails the task is performed by the redundant component, which reduces mortality. Like this, variability and adaptation are ensured even under environmental changes. In [49, 50] a multi-agent-system is proposed as a successful example for integrating degeneracy. The concept of degeneracy has two major drawbacks: First, multi-functional components requires more resources and are more complicated, which contradicts the goal of simple components in complex systems. Second, degenerate mappings are a contradictions to the mathematically well-behaved mappings (e.g. bijections) that are typically preferred in optimizations.

There exist different concepts for including robustness as a stability criterion in complex systems, either as an additional meta-attribute or in the evolvability definition. We intentionally avoid the stated conflict between robustness and variability, and instead include it in regularity and adaptation potential, the other two aspects of the proposed evolvability definition.

B. Modularity

Modularity is one important feature that has to be incorporated into complex systems as it increases their performance. The articles [11, 13, 42] link evolutionary biology and evolutionary computer science, and emphasize the importance of the genotype-phenotype mapping. Modularity is one property that can be included in this genotype-phenotype relation. In the biological context it describes the independence of functionally different regions of an individual. Independent components reduce mortality in case one component fails, and therefore induce robustness from another perspective as discussed before [10, 47, 51]. The flexibility to replace components improves the adaptation potential. As a consequence, a high modularity promotes the development of a system, such that it can be evolved, optimized, or adapted easier and faster. We incorporate modularity into the choice of the genotype-phenotype mapping instead of defining it as an evolvability-relevant attribute. For instance, in an automotive design scenario different shape morphing methods can be used, such that the optimization may pick the most promising one for different regions of the shape.

C. Further Targets of Evolvability

The evolutionary algorithm that is used for developing and optimizing engineering systems has a strong influence on the resulting performance. For instance, if it hinders variability the system cannot change or adapt. This is investigated in the field of genetic programming, e.g., in [52–55]. Evolvability, regarded as the adaptation potential of a population to the environment, characterizes the mutation, the recombination, and the selection criterion of evolutionary algorithms. Since these are stochastic algorithms, a probability-based definition of evolvability is commonly used. Obtaining the probability measures (e.g., the probability that a phenotype varies) is one major problem, and methods based on additional optimization steps or local information are frequently used. Of course, algorithmic operators, such as the adaptation of mutation step size or recombination probability, repair and support the evolvability of complex systems.

In the more general context—evolvable hardware—different targets of the quality criterion evolvability are investigated in [56–62]. In these works the setup and choice of algorithms is discussed, as well as the representation setup, or different fitness strategies to achieve different optimization goals. The focus is set to these aspects, instead of to a general quality criterion. Many relevant attributes of complex systems are covered, such as the use of simple components, modularity, evolutionary methods, or varying goals. This enriches our understanding of the different attributes that can be evaluated by evolvability.

V. Conclusion and Prospect

The most frequently used approach for solving engineering tasks still is *classical* engineering: dividing the problem into sub-problems, simplifying and solving them. After this process the solution is fixed, which prevents the adaptation to unpredictable environments. With growing problem and system complexity due to, e.g., nonlinear interactions of components, feedback loops, and adaptive processes during a system's lifetime, the unpredictability of system behavior further increases. Therefore the complex system engineering approach does not target specialized solutions, but rather aims to implement a blueprint and development guidelines to let the system unfold and adapt itself while interacting with the environment.

The representation setup significantly determines the development capability of a system. It has to be flexible enough for the system to be able to automatically adjust to changing objectives/environments during the system's lifetime. This level of self-organization, led by an evolutionary process, can result in beneficial unexpected solutions. The designer is responsible for the layout of the components or representation to enable a constant evaluation and adaptation of the representation.

In this paper we considered *evolvability* as a quality criterion for measuring the potential of a representation for complex systems. Based on basic analyses in biology, evolvability characterizes the potential success of a population in evolution. We analyzed concepts in biology and their transfer to technical systems, and pointed out disadvantages and promising approaches. Summarizing from literature a three-stage classification is motivated.

The first aspect that evolvability has to cover is a fitness-independent quality, which we interpret as *regularity*. It is included to prevent unfavorable designs. The second aspect preserves the potential variety, and is fitness-independent, too. If we achieved a beneficial design during the design optimization we need this *variability* to react on changing environments or targets. Some design regions may be crucial for the design process. It is important that the representation promotes the adaptation of these regions. Quantifying this *adaptation potential* improves the performance of the design process and is a third aspect of evolvability.

We discussed robustness in this context, since it is an important feature in engineering. The basic idea is to induce stability, reduce mortality, and thereby improve the evolutionary progress. It can be analyzed on the three levels like evolvability. With respect to variability contradicting positions exist in literature. We have shown arguments supporting the assumption that robustness promotes variability and arguments against this statement. We implicitly included robustness into our evolvability concept, since we regard a regular or adaptable individual as robust.

We are aware that the optimization algorithms for engineering complex systems influence their performance, too. In this survey, however, we focus mostly on the representation setup. Other aspects or more concrete definitions of regularity, variability, and adaptation potential (in a specific application context) could be analyzed in future studies. We gathered the articles that we included in this survey and categorize them in Table I. Short notices to the different approaches towards evolvability, their basic results, and the context of the evolvability analysis are summarized in this table. Our next step is to define evolvability precisely for an automotive design optimization scenario and to evaluate this definition through extensive simulations.

Acknowledgments

Andreas Richter gratefully acknowledges the financial support from Honda Research Institute Europe (HRI-EU). Mario Botsch is supported by the German National Research Foundation (DFG CoE 277: CITEC).

TABLE I. ASPECTS AND DEFINITIONS OF EVOLVABILITY

	Definitions	Result	Context	Articles
theoretical approaches	evolvability = variability and modularity	genotype-phenotype mapping is a key object for the representation problem; robustness, modularity, variation are important for evolvability; fitness independent definition	biological analysis, biological principles in internet evolution	[10, 11, 13, 42, 47]
	evolvability = variability and adaptation	changing fitness promotes variation thus evolvability; fitness dependent definition of evolvability	evolutionary computation	[63]
	evolvability = variability and heritability	heritability alone is not a good measure; evolvability definition is highly problem dependent; evolvability definition is fitness and selection criterion dependent	biological analysis of animal population studies	[15, 17, 18]
	evolvability as probabilistic model	evolvability is the probability of a future trait given the current environmental and features of the population; promoting features are (low) mutation rate and variation	biological analysis	[14]
		mathematical framework for algorithm analysis; structures (e.g. specific functions) are evolvable with respect to different probability distributions	conceptual mathematical analysis	[64–68]
	robustness as probabilistic model	mathematical (probability) framework of robustness for networks defined	mathematical biological networks analysis	[69–72]
	robustness = regularity and variation and adaptation	robustness is persistence of a trait under perturbations; perturbation are: noise, genetic variation, environmental variation	biological analysis	[19]
	cryptic gene variation as biological concept of robustness and adaptation	genotype variety of a phenotype promotes robustness in a stable environment and adaptation in a varying environment; cryptic gene variation depends on neutrality	biological analysis	[44]
	evolvability as a criterion for long term development	many attributes have to be incorporated, e.g., modularity, adaptation, testability, variability, consistency, etc.	software engineering	[73]
biological simulation	evolvability, robustness = variation on the genotype and phenotype	genotype robustness hinders genotype evolvability, phenotype robustness promotes phenotype evolvability; definitions based on neutrality; fitness independent	RNA, Transcription factor binding sides analysis, neutral networks	[12, 21, 25, 74]
	variation additionally defined through probability	robustness promotes this definition	gene regulatory circuits, protein simulations	[22–24, 43]
		recombination/mutation rate analysis; setup of the algorithm highly influences evolvability	algorithm analysis	[41, 75]
	evolvability as product of phenotype and genotype distance	robustness integrated in definition automatically; fitness independent	transcription factors in cell growth	[26]
	evolvability = adaptation	robustness-evolvability relation is non-monotonic	RNA simulations	[32]
	degeneracy = multi-functionality	switch between redundancy and diversity (dependent on the environment) improves robustness and evolvability (adaptation); concept based on neutrality neutrality	simulations of a multi-agent system	[7, 49, 50]
algorithm setup in engineering	novelty (variation) search as selection criterion in algorithms	positive correlation between novelty search and optimization performance; sampling needed to compute variation (evolvability)	maze navigation, robot walk	[28, 31, 76, 77]
		combination of novelty and fitness based search is most promising	neural networks in maze navigation, pattern guessing, robot walk	[29]
	evolvability as number of successful solved problems	novelty search worse than fitness search when target changes	maze navigation	[30]
	evolvability = performance of algorithm	modularity of genotype-phenotype mapping increases evolvability	maze navigation, neural networks	[51]
		gradient information used to improve individuals after recombination; fitness dependent	genetic programming, gene expression programming	[52, 53, 55]
		genotype size, choice and setup of algorithms investigated with respect to fast fitness improvement	evolvable hardware	[56–62]
evolvability for representations	evolvability = robustness and variation (and heritability)	increasing evolvability promotes optimization; variation and robustness negatively correlated; evolvability as tradeoff; heritability gained through control volume setup; fitness independent	continuous free-from deformation, discrete Boolean functions	[20, 27, 45, 46]
	evolvability = adaptation potential	evolvable setups superior to robust ones regarding the adaptation to new fitness environments; compromise between fitness dependent and independent definition most successful	spline matching, pattern guessing, neural network learning, sting writing grammar, hexapod simulation	[33–38, 40, 78]
		redundancy promotes evolvability	grammar evolution	[48]
		predicting the fitness development improves evolvability and optimization performance; model based on probability measures used for a search operator	car optimization, simulation on test functions	[39, 54]

REFERENCES

[1] D. Sieger, S. Menzel, and M. Botsch, "On shape deformation techniques for simulation-based design optimization," in *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, 2015, to appear.

[2] (2006) A consensus of the incose fellows. [Online]. Available: http://www.incose.org/practice/fellowsconsensus.aspx

[3] R. Frei and G. D. M. Serugendo, "Concepts in complexity engineering," *International Journal of Bio-Inspired Computation*, vol. 3, no. 2, pp. 123–139, 2011.

[4] M. Ulieru and R. Doursat, "Emergent engineering: a radical paradigm shift," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 4, no. 1, pp. 39–60, 2011.

[5] A. A. Mina, D. Braha, and Y. Bar-Yam, "Complex engineered systems: A new paradigm," in *Complex Engineered Systems*, ser. Understanding Complex Systems. Springer, 2006, pp. 1–21.

[6] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33, pp. 3190–3218, 2007.

[7] J. M. Whitacre, "Degeneracy: a link between evolvability, robustness and complexity in biological systems," *Theoretical Biology and Medical Modelling*, vol. 7, no. 1, pp. 1–17, 2010.

[8] A. B. Urken, A. B. Nimz, and T. M. Schuck, "Designing evolvable systems in a framework of robust, resilient and sustainable engineering analysis," *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 553–562, 2012.

[9] M. Onori, D. Semere, and B. Lindberg, "Evolvable systems: An approach to self-X production," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 5, pp. 506–516, 2011.

[10] M. Kirschner and J. Gerhart, "Evolvability," *Proceedings of the National Academy of Sciences*, vol. 95, no. 15, pp. 8420–8427, 1998.

[11] G. P. Wagner and L. Altenberg, "Perspectives: Complex adaptations and the evolution of evolvability," *Evolution*, vol. 50, no. 3, pp. 967–976, 1996.

[12] A. Wagner, "Robustness and evolvability: a paradox resolved," *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, no. 1630, pp. 91–100, 2008.

[13] M. Pigliucci, "Is evolvability evolvable?" *Nature Reviews Genetics*, vol. 9, no. 1, pp. 75–82, 2008.

[14] R. L. Brown, "What evolvability really is," *British Journal for the Philosophy of Science*, vol. 65, no. 3, 2013.

[15] J. F. Y. Brookfield, "Evolution and evolvability: celebrating Darwin 200," *Biology Letters*, vol. 5, no. 1, pp. 44–46, 2009.

[16] K. Sterelny, "Symbiosis, evolvability and modularity," *Modularity in development and evolution*, pp. 490–516, 2004.

[17] D. Houle, "Comparing evolvability and variability of quantitative traits," *Genetics*, vol. 130, no. 1, pp. 195–204, 1992.

[18] T. F. Hansen, C. Plabon, and D. Houle, "Heritability is not evolvability," *Evolutionary Biology*, vol. 38, no. 3, pp. 258–277, 2011.

[19] M.-A. Felix and A. Wagner, "Robustness and evolution: concepts, insights and challenges from a developmental model system," *Heredity*, vol. 100, no. 2, pp. 132–140, 2006.

[20] H. Lehmann and S. Menzel, "Evolvability as concept for the optimal design of free-form deformation control volumes," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[21] A. Wagner, "Robustness, evolvability, and neutrality," *FEBS Letters*, vol. 579, no. 8, pp. 1772–1778, 2005.

[22] J. L. Payne, J. H. Moore, and A. Wagner, "Robustness, evolvability, and the logic of genetic regulation," *Artificial Life*, vol. 20, no. 1, pp. 111–126, 2014.

[23] J. L. Payne and J. H. Moore, "Robustness, evolvability, and accessibility in the signal-integration space of gene regulatory circuits," in *Proceedings of the European Conference on Artificial Life (ECAL)*, 2011, pp. 662–669.

[24] Y. Wang, J. Bryson, and S. Matthews, "Evolving evolvability in the context of environmental change: A gene regulatory network (grn) approach," in *Artificial Life: Proceedings of the International Conference on the Synthesis and Simulation of Living Systems*. The MIT Press, 2014.

[25] J. L. Payne and A. Wagner, "The robustness and evolvability of transcription factor binding sites," *Science*, vol. 343, no. 6173, pp. 875–877, 2014.

[26] Y. Jin and J. Trommler, "A fitness-independent evolvability measure for evolutionary developmental systems," in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2010, pp. 1–8.

[27] S. Menzel, "Evolvable free-form deformation control volumes for evolutionary design optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 1388–1395.

[28] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011.

[29] B. Inden, Y. Jin, R. Haschke, H. Ritter, and B. Sendhoff, "An examination of different fitness and novelty based selection methods for the evolution of neural networks," *Soft Computing*, vol. 17, no. 5, pp. 753–767, 2013.

[30] D. Shorten and G. Nitschke, "How evolvable is novelty search?" 2014. [Online]. Available: http://pubs.cs.uct.ac.za/archive/00000974/01/ShortenNitschke-SSCI-2014.pdf

[31] R. Velez and J. Clune, "Novelty search creates robots with general skills for exploration," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*. ACM, 2014, pp. 737–744.

[32] J. A. Draghi, T. L. Parsons, G. P. Wagner, and J. B. Plotkin, "Mutational robustness can facilitate adaptation," *Nature*, vol. 463, no. 7279, pp. 353–355, 2010.

[33] H. Suzuki, "An example of design optimization for high evolvability: string rewriting grammar," *Biosystems*, vol. 69, no. 2, pp. 211–221, 2003.

[34] M. Bercachi, P. Collard, M. Clergue, and S. Verel, "Do not choose representation just change: An experimental study in states based ea," in *Proceedings of Annual Conference on Genetic and Evolutionary Computation (GECCO)*. ACM, 2009, pp. 1799–1800.

[35] D. Tarapore and J.-B. Mouret, "Comparing the evolvability of generative encoding schemes," in *Artificial Life: Proceedings of the International Conference on the Synthesis and Simulation of Living Systems*. MIT Press, 2014, pp. 55–62.

[36] J. Reisinger, K. O. Stanley, and R. Miikkulainen, "Towards an empirical measure of evolvability," in *Proceedings of workshop on Genetic and evolutionary computation*. ACM Press, 2005, pp. 257–264.

[37] J. Reisinger and R. Miikkulainen, "Selecting for evolvable representations," in *Proceedings of annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 1297–1304.

[38] J. Reisinger and R. Miikkulainen, "Acquiring evolvability through adaptive representations," in *Proceedings of annual conference on Genetic and evolutionary computation*, 2007, pp. 1045–1052.

[39] M. N. Le, Y. S. Ongand, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by adapting surrogates," *Evolutionary computation*, vol. 21, no. 2, pp. 313–340, 2013.

[40] N. Aulig, "Evolutionary optimization of the robustness and evolvability of design solutions," Master's thesis, Technische Universität Darmstadt, 2011.

[41] J. Masel and M. V. Trotter, "Robustness and evolvability," *Trends in Genetics*, vol. 26, no. 9, pp. 406–414, 2010.

[42] T. Hu and W. Banzhaf, "Evolvability and speed of evolutionary algorithms in light of recent developments in biology," *Journal of Artificial Evolution and Applications*, vol. 2010, pp. 1–28, 2010.

[43] J. D. Bloom, S. T. Labthavikul, C. R. Otey, and F. H. Arnold, "Protein stability promotes evolvability," *Proceedings of the National Academy of Sciences*, vol. 103, no. 15, pp. 5869–5874, 2006.

[44] J. M. Whitacre and S. P. Atamas, "The diversity paradox: How nature resolves an evolutionary dilemma," *arXiv preprint arXiv:1112.3115*, 2011.

[45] Y. Jin, R. Gruna, and B. Sendhoff, "Pareto analysis of evolutionary and learning systems," *Frontiers of Computer Science in China*, vol. 3, no. 1, pp. 4–17, 2009.

[46] Y. Jin, R. Gruna, I. Paenke, and B. Sendhoff, "Evolutionary multi-objective optimization of robustness and innovation in redundant genetic representations," in *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*. IEEE, 2009, pp. 38–45.

[47] C. Dovrolis and J. T. Streelman, “Evolvable network architectures: What can we learn from biology?” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 72–77, 2010.

[48] M. H. Luerssen and D. M. W. Powers, “Evolvability and redundancy in shared grammar evolution,” in *Proceedings of IEEE Congress on Evolutionary Computation*, 2007, pp. 370–377.

[49] J. M. Whitacre, P. Rohlfshagen, A. Bender, and X. Yao, “Evolutionary mechanics: new engineering principles for the emergence of flexibility in a dynamic and uncertain world,” *Natural computing*, vol. 11, no. 3, pp. 431–448, 2012.

[50] J. M. Whitacre, P. Rohlfshagen, A. Bender, and X. Yao, “The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments,” in *Proceedings of Parallel Problem Solving from Nature*, 2010, pp. 284–293.

[51] P. Durr, C. Mattiussi, and D. Floreano, “Genetic representation and evolvability of modular neural controllers,” *IEEE Computational Intelligence Magazine*, vol. 5, no. 3, pp. 10–19, 2010.

[52] L. Altenberg, “The evolution of evolvability in genetic programming,” *Advances in genetic programming*, vol. 3, pp. 47–74, 1994.

[53] N. P. A. Browne and M. V. dos Santos, “Adaptive representations for improving evolvability, parameter control, and parallelization of gene expression programming,” *Applied Computational Intelligence and Soft Computing*, vol. 2010, 2010.

[54] J. A. S. Alcaraz, G. Ochoa, M. Carpio, and H. Puga, “Evolvability metrics in adaptive operator selection,” in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*. ACM, 2014, pp. 1327–1334.

[55] B. Burlacu, M. Affenzeller, and M. Kommenda, “On the evolutionary behavior of genetic programming with constants optimization,” in *Computer Aided Systems Theory - EUROCAST 2013*, ser. Lecture Notes in Computer Science, R. Moreno-Daz, F. Pichler, and A. Quesada-Arencibia, Eds. Springer, 2013, vol. 8111, pp. 284–291.

[56] T. G. Kalganova, “Evolvable hardware design of combinational logic curcuits,” Ph.D. dissertation, Napier University Edinburgh, Scotland, 2000.

[57] J. Torresen, “An evolvable hardware tutorial,” in *Field Programmable Logic and Application*, ser. Lecture Notes in Computer Science, J. Becker, M. Platzner, and S. Vernalde, Eds. Springer, 2004, pp. 821–830.

[58] E. Stomeo, T. Kalganova, and C. Lambert, “Analysis of genotype size for an evolvable hardware system,” in *Proceedings of International Enformatika Conference Prague*, 2005, pp. 74–79.

[59] A. T. Soto, E. E. P. de Leon-Senti, A. H. Aguirre, M. D. T. Soto, and E. Daz-Daz, “A robust evolvable system for the synthesis of analog circuits,” *Computacin y Sistemas*, vol. 13, no. 4, pp. 409–421, 2010.

[60] P. C. Haddow and A. M. Tyrrell, “Challenges of evolvable hardware: past, present and the path to a promising future,” *Genetic Programming and Evolvable Machines*, vol. 12, no. 3, pp. 183–215, 2011.

[61] P. Rustem, *Genetic Algorithms: An Overview with Applications in Evolvable Hardware*, D. S. G. (Ed.), Ed. InTech Open Access Publisher, 2012.

[62] A. K. Srivastava, “Novel crossover for evolvable hardware,” *World Applied Sciences Journal*, vol. 21, no. 10, pp. 1408–1414, 2013.

[63] M. R. Glickman and K. Sycara, “Evolvability and static vs. dynamic fitness,” in *Workshop Proceedings of Artificial Life VII*, 2000.

[64] E. Angelino and V. Kanade, “Attribute-efficient evolvability of linear functions,” in *Proceedings of the 5th conference on Innovations in theoretical computer science*, 2014, pp. 287–300.

[65] L. Valiant, “Evolvability,” in *Mathematical Foundations of Computer Science 2007*. Springer, 2007, pp. 22–43.

[66] P. Valiant, “Distribution free evolvability of polynomial functions over all convex loss functions,” in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 142–148.

[67] V. Feldman, “Robustness of evolvability,” in *Proceedings of COLT*, 2009, pp. 277–292.

[68] M. Toussaint, “On the evolution of phenotypic exploration distributions,” in *Proceedings of Foundations of Genetic Algorithms 7 (FOGA VII)*. Morgan Kaufmann, 2003, pp. 169–182.

[69] B.-S. Chen and Y.-P. Lin, “On the interplay between the evolvability and network robustness in an evolutionary biological network: A systems biology approach,” *Evolutionary Bioinformatics*, vol. 7, pp. 201–233, 2011.

[70] B.-S. Chen and Y.-P. Lin, “A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their trade-off on phenotype robustness in biological networks part i: Gene regulatory networks in systems and evolutionary biology,” *Evolutionary Bioinformatics*, vol. 9, pp. 43–68, 2013.

[71] B.-S. Chen and Y.-P. Lin, “A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their tradeoff on phenotype robustness in biological networks part ii: Ecological networks,” *Evolutionary Bioinformatics*, vol. 9, pp. 69–85, 2013.

[72] B.-S. Chen and Y.-P. Lin, “A unifying mathematical framework for genetic robustness, environmental robustness, network robustness and their trade-offs on phenotype robustness in biological networks. part iii: Synthetic gene networks in synthetic biology,” *Evolutionary Bioinformatics*, vol. 9, pp. 87–109, 2013.

[73] R. Brcina, S. Bode, and M. Riebisch, “Optimisation process for maintaining evolvability during software evolution,” in *Proceedings of IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, 2009, pp. 196–205.

[74] S. F. Greenbury, I. G. Johnston, A. A. Louis, and S. E. Ahnert, “A tractable genotype-phenotype map for the self-assembly of protein quaternary structure,” *arXiv preprint arXiv:1311.0399*, 2013.

[75] M. A. Bedau and N. H. Packard, “Evolution of evolvability via adaptation of mutation rates,” *Biosystems*, vol. 69, no. 2, pp. 143–162, 2003.

[76] J. Lehman and K. O. Stanley, “Improving evolvability through novelty search and self-adaptation,” in *Proceedings of IEEE Congress on Evolutionary Computation*. IEEE, 2011, pp. 2693–2700.

[77] J. Lehman and K. O. Stanley, “Evolvability is inevitable: Increasing evolvability without the pressure to adapt,” *PloS one*, vol. 8, no. 4, 2013.

[78] J. Kim, D. Vandamme, J.-R. Kim, A. G. Munoz, W. Kolch, and K.-H. Cho, “Robustness and evolvability of the human signaling network,” *PLoS Computational Biology*, vol. 10, 2014.