Chapter 3
Evaluation of Complexity in Product Development

The term "complexity" stems from the Latin word "complexitas," which means comprehensive or inclusive. In current usage, it is the opposite of simplicity, though this interpretation does not appear to be underpinned by any explicit concept that could be directly used for the development of scientifically rigorous models or metrics. Various disciplines have studied the concepts and principles of complexity in basic and applied scientific research. Several frameworks, theories and measures have been developed, reflecting the differing views of complexity between disciplines. An objective evaluation of structural and dynamic complexity in PD would benefit project managers, developers and customers alike, because it would enable them to compare and optimize different systems in analytical and experimental studies. To obtain a comprehensive view of organizational, process and product elements and their interactions in the product development environment, a thorough review of the notion of complexity has to start from organizational theory (Section 3.1). The literature on organizational theory shows that the complexity of PD projects results from different "sources" and the consideration of the underlying organizational factors and their interrelationships is essential to successful project management (Kim and Wilemon 2009). However, our analyses have shown that static factor-based approaches are not sufficient to evaluate emergent complexity in open organizational systems and therefore the complexity theories and measures of basic scientific research must also be taken into account to capture the inherently complex nature of the product development flow (cf. Amaral and Uzzi 2007). These theories and measures can provide deeper insights into emergent phenomena of complex sociotechnical systems and dynamic mechanisms of cooperation (Section 3.2). Selected measures can also be used to optimize the project organization (Schlick et al. 2009, see Sections 5.2 and 5.3). The measures build upon our intuitive assessment that a system is complex if it is difficult to describe. The description can focus on structure, processes or both. In the description not only the length and the format are relevant but also the expressive power of the "description language." Furthermore, in a process-centered view, for many nontrivial systems the difficulty of prediction and retrodiction have to be

C. Schlick, B. Demissie, *Product Development Projects*, Understanding Complex Systems, DOI 10.1007/978-3-319-21717-8_3

simultaneously taken into account to obtain valid results. Comprehensive overviews of this and related concepts including detailed mathematical analyses and illustrations can be found in Shalizi (2006), Prokopenko et al. (2009) and Nicolis and Nicolis (2007). We will describe the main concepts and methods of basic scientific research in Section 3.2 based on the material from Shalizi (2006). For effective complexity management in PD, the product-oriented measures from theories of systematic engineering design are also relevant (Section 3.3). Seminal work in this field has been done by Suh (2005) on the basis of information-theoretic quantities. These quantities are also the foundation of statistical complexity measures from basic scientific research, which means that Suh's complexity theory and recent extensions of it (see Summers and Shah 2010) must be discussed in the light of the latest theoretical developments. Moreover, the literature that has been published concerning the design structure matrix (Steward 1981) as a universal dependency modeling technique has to be considered (see e.g. Lindemann et al. 2009; Eppinger and Browning 2012). This literature also provides a firm foundation for quantitative modeling of cooperative work in PD projects by means of either time- or task-based design structure matrices (see e.g. Gebala and Eppinger 1991; Smith and Eppinger 1997; Schlick et al. 2007). In general, we have sought to restrict our analyses to mature scientific theories because of their universality, objectivity and validity.

3.1 Approaches from Organizational Theory

According to Murmann (1994) and Griffin (1997), complexity in the product development environment is determined by the number of (different) parts in the product and the number of embodied product functions. This basic approach can be used to assess complexity in different types of PD projects, for instance the five classic types defined by Wheelwright and Clark (1992): research and development, breakthrough, platform, derivative, and alliances and partnership projects. To make this approach fully operational, Kim and Wilemon (2003) developed a complexity assessment template covering these and other important "sources." The first source in their assessment template is "technological complexity," which can be divided into "component integration" and "technological newness." The second source is the "market (environmental) complexity" that results from the sensitivity of the project's attributes to market changes. "Development complexity" is the third source and is generated when different design decisions and components have to be integrated, qualified suppliers have to be found and supply chain relationships have to be managed. The fourth source is "marketing complexity," which results from the challenges of bringing the product to market. "Organizational complexity" is the fifth source, because projects usually require intensive cooperation and involve many areas of the firm. Their coordination leads to "intraorganizational complexity," the sixth source. When in large-scale engineering projects many other companies such as highly specialized engineering service providers are involved

and must be coordinated in a continuous integration rhythm, this source should be extended and cover both inter- and intraorganizational complexity. In order to validate and prioritize sources of complexity, Kim and Wilemon (2009) conducted an extensive empirical investigation. An analysis of exploratory field interviews with 32 project leaders and team members showed that technological challenges, product concept/customer requirement ambiguities and organizational complexity are major issues that generate complexity in PD. The perceived dominant source was technological challenges, since roughly half of the respondents noted technological difficulties encountered in attempting to develop a product using an unproven technique or process. With regard to complexity in the management of projects of different types—not necessarily focusing on (new) product development projects—Mulenburg (2008) distinguishes between the following six sources: (1) Details: number of variables and interfaces, (2) Ambiguity: lack of awareness of events and causality, (3) Uncertainty: inability to pre-evaluate actions, (4) Unpredictability: inability to know what will happen, (5) Dynamics: rapid rate of change, and (6) Social structure: number and types of interactions between actors.

Hölttä-Otto and Magee (2006) developed a project complexity framework based on the seminal work of Summers and Shah (2003). They identified three dimensions: the product itself (artifact), the project mission (design problem), and the tasks required to develop the product (process). The key indicators for each of these dimensions are size, interactions and stretch (solvability). Hölttä-Otto and Magee conducted interviews in five divisions of large corporations competing in different industries on the North American market. Their findings show that the effort estimation is primarily based on the scale and the stretch of the project. Surprisingly, they found no utilization of the level of either component or task interactions in estimating project complexity. Further, they found no empirical evidence for interactions being a determinant of project difficulty (Hölttä-Otto and Magee 2006). Tatikonda and Rosenthal (2000) focus on the task dimension and relate project complexity to the nature, quantity and magnitude of the organizational subtasks and subtask interactions required by a project.

A recent work combining a literature review and their own empirical work on the elements that contribute to complexity in large engineering projects was published by Bosch-Rekveldt et al. (2011). The analysis of the literature sources and 18 semi-structured interviews in which six completed projects were studied in depth led to the development of the TOE framework. The framework covers 50 different elements, which are grouped into three main categories: “technical complexity” (T), “organizational complexity” (O) and “environmental complexity” (E). Additional subcategories of TOE are defined on a lower level: “goals,” “scope,” tasks,” “experience,” “size,” “resources,” “project team,” “trust,” “stakeholders,” “location,” “market conditions,” and “risks,” showing that organizational and environmental complexity are more often linked with softer, qualitative aspects. Interestingly, Bosch-Rekveldt et al. (2011) distinguish between project complexity and project management (or managerial) complexity. Project management complexity is seen as a subset of project complexity. Various normative organizing

principles for coping with managerial complexity can be found in the standard literature on project management (e.g. Shtub et al. 2004; Kerzner 2009). If, for instance, the level of managerial complexity is low, project management within the classic functional organizational units of the company is usually most efficient and cross-functional types of project organization can create unnecessary overhead. However, if coordination needs between functional, spatial and temporal boundaries are high, a matrix organization is often the better choice, as it allows development projects to be staffed with specialists from across the organization (Shtub et al. 2004). The preferred organizational structure for large-scale, long-term engineering projects is pure project organization. The inherent advantage of this type of structure is that responsibilities for the project lie with one team, which works full-time on the project tasks throughout the entire project life cycle. Specific sources of managerial complexity and their impact on performance were also examined in the literature, e.g. communication across functional boundaries (Carlile 2002), cross-boundary coordination (Kellogg et al. 2006), spatial and temporal boundaries in globally distributed projects (Cummings et al. 2009), and the effects of a misalignment in the geographic configuration of globally distributed teams (O'Leary and Mortensen 2010). Maylor et al. (2008) developed an integrative model of perceived managerial complexity in project-based operations. Based on a multistage empirical study elements of complexity were identified and classified under the dimensions of "mission," "organization," "delivery," "stakeholder," and "team."

The literature review shows that there are a large variety of nomenclatures and definitions for the sources of complexity in PD projects. However, the underlying factors have not yet been integrated into a single objective and valid framework. According to Lebcir (2011) there is an urgent need for a new, non-confusing, and comprehensive framework that is derived from the extensive body of available knowledge. He suggests a framework in which "project complexity" is decomposed into "product complexity" and "innovation." Product complexity refers to structural complexity (see Section 3.3) and is determined by "product size" in terms of the number of elements (components, parts, subsystems, functions) in the product and by "product interconnectivity," which represents the level of linkages between elements. On the other hand, innovation refers to "product newness" and "project uncertainty." Product newness represents the degree of redesign of the product compared to previous generations of the same or similar products. Project uncertainty represents the fact that methods and capabilities are often not clearly defined at the start of a project. The results of a dynamic simulation indicate that an increase in uncertainty has a significant impact on the development time. The other factors also tend to increase development time as they increase, but their impact is not significantly different in projects involving medium or high levels of these factors.

In reviews of the more practice-oriented project management literature, two complexity models have received considerable attention, especially at large-scale development organizations: (1) the UCP (uncertainty, complexity and pace) model and the (2) NTCP (novelty, technology, complexity and pace) model. Both models were developed by Shenhar and colleagues (Shenhar and Dvir 1996, 2007; Shenhar

1998). In principle, these models can be applied to all types of projects. The UCP model is based on a conceptual two-dimensional taxonomy which classifies a project according to four levels of technological uncertainty and three levels of system scope. The four levels of technological uncertainty (low-tech, medium-tech, high-tech and super-high-tech) mainly refer to the uncertainty as perceived by the organization at the time of the project's initiation and thereby indicate how soon the product functions can be concretized. Moreover, they characterize the extent of new and therefore possibly premature technologies that are needed to reach the project goals. In the UCP model, the second dimension of system scope is based on the complexity of the system as expressed by the different hierarchies inside the product (assembly, system and array). Since systems are composed of subsystems and subsystems of components, hierarchies usually involve many levels. Hierarchies apply to systems as well as to tasks, which together determine the overall complexity of the project. The element of time in terms of "pace" was added to the model to account for the urgency and criticality of reaching milestones, as milestones with different time constraints call for different managerial strategies (Dvir et al. 2006). When complexity, uncertainty or pace increase, project planning becomes more difficult and the risk of project failure increases. Consequently, the formality of project management must also increase. The UCP model is based on quantitative and qualitative analyses of more than 250 projects within the US and Israeli defense and industry sectors. However, since it is usually used in retrospect rather than at the outset of new projects, the UCP model has a descriptive character. In contrast, the NTCP model, also called the "Diamond Framework," was developed as a prescriptive model in order to analyze projects and provide a better understanding of what needs to be done in order to ensure their success. The Diamond Framework is based on four pillars. The first pillar, "novelty," refers to the degree of newness (derivate vs. platform vs. breakthrough) of project results or crucial aspects of the project. With varying degrees of novelty, different requirements must be satisfied and corresponding action plans have to be developed. The second pillar represents the level of "technological uncertainty" in a project. Technological uncertainty is primarily determined by the level of new and mature technology required (low vs. medium vs. high vs. super-high technology). As the level of technological uncertainty rises, the risk of failure and efficiency loss increases. "Complexity," the third pillar, describes the types of arrangement between elements within the system, especially their hierarchical structure (assembly vs. system vs. array). Higher degrees of complexity entail more interaction between elements, which in turn demands higher project management formality. The fourth and final pillar of the NTCP model is "pace." As within the UCP model, pace refers to the urgency of reaching time goals and milestones. It chiefly depends on the available time for project completion and is divided into four types: regular, fast/competitive, time-critical and blitz. By considering all four pillars of the NTCP model at the beginning of the project and revisiting them as it progresses, project managers are provided with a methodology for assessing the uniqueness of their project and selecting appropriate management methods and techniques for coping with complexity.

Like the UCP and NCTP models, the Project Complexity Model developed by Hass (2009) has received a great deal of attention in the practice-oriented project management community. This model offers a broad framework for identifying and diagnosing the aspects of complexity within a project so that the project team can make appropriate management decisions. The model captures a number of sources of project complexity, including project duration and value; team size and composition; urgency; schedule, cost, and scope flexibility; clarity of the problem and solution; stability of requirements; strategic importance; stakeholder influence; level of organizational and commercial change; external constraints and dependencies; political sensitivity; and unproven technology (Hass 2009). The detailed complexity dimensions are shown in Table 3.1. The Project Complexity Model can also be used to evaluate the complexity of a particular project in an enterprise. To carry out the evaluation, Hass (2009) developed a corresponding "Project Complexity Formula," which is summarized in Table 3.2.

The complexity templates and frameworks that have been developed in organization theory and neighboring disciplines are especially beneficial for the management of product development projects because they help to focus managerial intervention on empirically validated performance-shaping factors and key elements of complexity. It must be criticized, though, that without a quantitative theory of emergent complexity it is almost impossible to identify the essential variables and their interrelationships. Furthermore, it is very difficult to consolidate them into one consistent complexity metric. In the literature very few authors, such as Mihm et al. (2003, 2010), Rivkin and Siggelkow (2003, 2007), and Braha and Bar-Yam (2007) build upon quantitative scientific concepts for the analysis of complex sociotechnical systems. Mihm et al. (2003) present analytical results from random matrix theory predicting that the larger the project, as measured by components or interdependencies, the more likely are problem-solving oscillations are and the more severe they become—failure rates grow exponentially. In the work of Rivkin and Siggelkow (2003, 2007), Kaufman's the famous biological evolution theory and the NK model are used to study organizations as systems of interacting decisions. Different interaction patterns such as block diagonal, hierarchical, scale-free, and so on are integrated into a simulation model to identify local optima. The results show that, by keeping the total number of interactions between decisions fixed, a shift in the pattern can alter the number of local optima by more than one order of magnitude. In a similar fashion Mihm et al. (2010) use a statistical model and Monte Carlo experiments to explore the effect of an organizational hierarchy on search solution stability, quality and speed. Their results show that assigning a lead function to "anchor" a solution speeds up problem-solving, that the choice of local solutions should be delegated to the lowest hierarchical level, and that organizational structure is comparatively unimportant at the middle management level, but does indeed matter at the "front line," where groups should be kept small. Braha and Bar-Yam (2007) examine the statistical properties of networks of people engaged in distributed development and discuss their significance. The autoregression models of cooperative work that were introduced in Chapter 2 (Eq. 8 and 39) are quite closely related to their dynamical model. However, there

Table 3.1 Complexity dimensions and project complexity profiles of the Project Complexity Model developed by Hass (2009)

Complexity dimensions	Project complexity profile		
	Independent	Moderately complex	Highly complex
Time/Cost	<3 months < $250K	3–6 months $250K–$750K	>6 months > $750K
Team Size	3–4 team members	5–10 team members	>10 team members
Team Composition and Performance	• Strong project leadership • Team staffed internally, has worked together in the past, and has a track record of reliable estimates • Formal, proven PM, BA and SE methodology with QA and QC processes defined and operational	• Competent project leadership • Team staffed with internal and external resources; internal staff has worked together in the past and has track record of reliable estimates • Contract for external resources is straightforward; contractor performance is known • Semi-formal methodology with QA/QC processes defined	• Project manager inexperienced in leading complex projects • Complex team structure of varying competencies (e.g., contractor, virtual, culturally diverse, outsourced) • Complex contracts; contractor performance unknown • Diverse methodologies
Urgency and Flexibility of Cost, Time and Scope	• Minimized scope • Small milestones • Flexible schedule, budget and scope	• Schedule, budget and scope can undergo minor variations, but deadlines are firm • Achievable scope and milestones	• Over-ambitious schedule and scope • Deadline is aggressive, fixed, and cannot be changed • Budget, scope and quality leave no room for flexibility
Clarity of Problem, Opportunity and Solution	• Clear business objectives • Easily understood problem, opportunity or solution	• Defined business objectives • Problem or opportunity is partially defined • Solution is partially defined	• Unclear business objectives • Problem or opportunity is ambiguous and undefined • Solution is difficult to define
Requirements Volatility and Risk	• Strong customer/user support • Basic requirements are understood, straightforward and stable	• Adequate customer/user support • Basic requirements are understood but are expected to change • Moderately complex functionality	• Inadequate customer/user support • Requirements are poorly understood, volatile and largely undefined • Highly complex functionality
Strategic Importance, Political Implications, Multiple Stakeholders	• Strong executive support • No political implications • Straightforward communications	• Adequate executive support • Some direct impact on mission • Minor political implications	• Mixed/inadequate executive support • Impact on core mission • Major political implications

(continued)

Table 3.1 (continued)

Complexity dimensions	Project complexity profile		
	Independent	Moderately complex	Highly complex
		• 2–3 stakeholder groups • Challenging communication and coordination effort	• Visible at highest levels of the organization • Multiple stakeholder groups with conflicting expectations
Level of Organizational Change	• Impacts a single business unit, one familiar business process and one IT system	• Impacts 2–3 somewhat familiar business units, processes and IT systems	• Large-scale organizational change that impacts the enterprise • Spans functional groups or agencies • Shifts or transforms the organization • Impacts many business processes and IT systems
Level of Commercial Change	• Minor changes to existing commercial practices	• Enhancements to existing commercial practices	• Groundbreaking commercial practices
Risks, Dependencies, and External Constraints	• Considered low risk • Some external influences • No challenging integration issues • No new or unfamiliar regulatory requirements • No punitive exposure	• Considered moderate risk • Some project objectives are dependent on external factors • Challenging integration effort • Some new regulatory requirements • Acceptable exposure	• Considered high risk • Overall project success largely depends on external factors • Significant integration required • Highly regulated or novel sector • Significant exposure
Level of IT Complexity	• Solution is readily achievable using existing, well-understood technologies • IT complexity is low	• Solution is difficult to achieve or technology is proven but new to the organization • IT complexity and legacy integration are moderate	• Solution requires groundbreaking innovation • Solution is likely to use immature, unproven or complex technologies provided by outside vendors • IT complexity and legacy integration are high

Table 3.2 Decision table of the Project Complexity Formula developed by Hass (2009)

Highly Complex	Moderately Complex	Independent
Level of change = large-scale enterprise impacts *or* Both the problem and the solution are difficult to define or understand, and the solution is difficult to achieve. The solution is likely to use unproven technologies. *or* Four or more categories in the "highly complex" column	Two or more categories in the "moderately complex" column *or* One category in the "highly complex" column and three or more in the "moderately complex" column	No more than one category in the "moderately complex" column *and* No categories in the "highly complex" column

To evaluate the complexity of a particular project, the boxes in the Project Complexity Model from Table 3.1 that best describe the project must be shaded out. Then, the complexity formula can be applied by following the decision rules above

are important differences: the VAR(1) models are defined over a continuous range of state values and can therefore represent different kinds of cooperative relationships as well as precedence relations (e.g. overlapping); each task is unequally influenced by other tasks; and finally, correlations ρ_{ij} between performance fluctuations among tasks i and j can be captured.

3.2 Approaches from Basic Scientific Research

3.2.1 Algorithmic Complexity

Historically, the most important measure from basic scientific research is algorithmic complexity, which dates back to the great mathematicians Kolmogorov, Solomonoff and Chaitin. They independently developed a measure known today as the "Kolmogorov−Chaitin complexity" (Chaitin 1987; Li and Vitányi 1997). In terms of information processing, the complexity of the intricate mechanisms of a nontrivial system can be evaluated using output signals, signs and symbols that are communicated to an intelligent observer. In this sense, complexity is manifested to an observer through the complicated way in which events unfold in time and are organized in state space. According to Nicolis and Nicolis (2007), the characteristic hallmarks of such spatiotemporal complexity are nonrepetitiveness, a pronounced variability extending over many scales of place and time, and sensitivity to initial conditions and to the other parameters. Furthermore, a given system can generate a variety of dependencies of this kind associated with the different states simultaneously available. If the transmitted output of a complex system is symbolic, it can be concatenated in the form of a data string x and may be sequentially stored in a computer file for post-hoc analysis. The symbols are typically chosen from a

predefined alphabet $\mathcal{X}$. If the output is a time- or space-continuous signal, it can be effectively encoded with methods of symbolic dynamics (Lind and Marcus 1995; Nicolis and Nicolis 2007). The central idea put forward by Kolmogorov, Solomonoff and Chaitin is that a generated string is "complex" if it is difficult for the observer to describe. The observer can describe the string by writing a computer program that reproduces it. The difficulty of description is measured by the length of the computer program on a Universal Turing Machine U. If x is transformed into binary form, the algorithmic complexity of x, termed $K_U(x)$, is the length of the shortest program with respect to U that will print x and then halt. According to Chaitin (1987), an additional requirement is that the string x has to be encoded by a prefix code $d(x)$. A prefix code is a type of code system that has no valid code word that is a prefix (start substring) of any other valid code word in the set. The corresponding universal prefix computer U has the property that if it is defined for a string s, then $U(st)$ is undefined for every string t that is not the empty string ε (Li and Vitányi 1997). The complete definition of the Kolmogorov−Chaitin complexity is:

$$K_U(x) = \min\{|d(p)| : U(p) = x\}. \tag{200}$$

In this sense, $K_U(x)$ is a measure of the computational resources needed to specify the data string x in the language of U. We can directly apply this algorithmic complexity concept to project management by breaking down the total amount of work involved in the project into fine-grained activities a_i and labeling the activities unambiguously by using discrete events e_i from a predefined set $\mathcal{X}\,(i = 1, \ldots, |\mathcal{X}|)$. During project execution it is recorded when activity a_i is successfully completed and this is indicated by scheduling the corresponding event e_i. The sequence of scheduled events $x = \left(e_{j(o)}, e_{j(1)} \ldots\right)\ \left(e_{j(i)} \in \mathcal{X}, j(\tau) \in \{1, \ldots, |\mathcal{X}|\}, \tau = 0, 1, \ldots\right)$ encodes how the events unfold in time and are organized in a goal-directed workflow. The index $j(\tau)$ can be interpreted as a pointer to the event e that occurred at position τ in the data sequence x. It is evident that a simple periodic work process whose activities are processed in strict cycles, like in an assembly line, is not complex because we can store a sample of the period and write a program that repeatedly outputs it. At the opposite end of the complexity range in the algorithmic sense, a completely unpredictable work process without purposeful internal organization cannot be described in any meaningful way except by storing every feature of task processing, because we cannot identify any persisting structure that could offer a shorter description. This example quite clearly shows that the algorithmic complexity is not a good measure for emergent complexity in PD projects, because it is maximal in the case of purely random task processing. Intuitively, such a state of "amnesia," in which no piece of information from the project history is valuable for improving the forecasts of the project manager and the team members, is not truly complex. Nor can the algorithmic complexity reveal the important long-range interactions between tasks or evaluate multilayer interactions in the hierarchy of an organization either. An additional conceptual weakness of the algorithmic complexity measure and its later refinements is that it aims for an exact description of

patterns. Many of the details of any configuration are simply random fluctuations from different sources such as human performance variability. Clearly, it is impossible to identify regularities from random fluctuations that generalize to other datasets from the same complex system; to assess complexity, the focus must be on the underlying regularities and rules shaping system dynamics. These regularities and rules must be distinguished from noise by employing specific selection principles. Therefore, a statistical representation is necessary that refers not to individual patterns but to a joint ensemble generated by a complex system in terms of an information source. In complex systems, the deterministic and probabilistic dimensions become two facets of the same reality: the limited predictability of complex systems (in the sense of the traditional description of phenomena) necessitates adopting an alternative view, and the probabilistic description allows us to sort out regularities of a new kind. On the other hand, far from being applied in a heuristic manner, in which observations have to fit certain preexisting laws imported from classical statistics, the probabilistic description we are dealing with here is "intrinsic" (Nicolis and Nicolis 2007), meaning that it is self-generated by the underlying system dynamics. Depending on the scale of the phenomenon, a complex system may have to develop mechanisms for controlling randomness to sustain a global behavioral pattern or, in contrast, to thrive on randomness and to acquire in a transient manner the variability and flexibility needed for its evolution between two such configurations. In addition to these significant conceptual weaknesses, a fundamental computational problem is that $K_U(x)$ cannot be calculated exactly. We can only approximate it "from above," which is the subject of the famous Chaitin theorem (Chaitin 1987). Later extensions of the classic concept of algorithmic complexity focus on complementary computational resources. In Bennett's (1988) logical depth the number of computing steps is counted that the minimum length program on a Universal Turing Machine U requires to generate the data string x. In Koppel and Atlan's (1991) theory of "sophistication" only the length of the part of the program on U is evaluated that captures all regularities of the data string. This means that, as with effective complexity (Gell-Mann 1995; Gell-Mann and Lloyd 1996; Gell-Mann and Lloyd 2004, see Section 3.2.3), irreducible random fluctuations that do not generalize to other datasets are sorted out. As with the Kolmogorov−Chaitin complexity, logical depth and sophistication are not computable, even with a generative model (Crutchfield and Marzen 2015).

3.2.2 Stochastic Complexity

The most prominent statistical complexity measure is Rissanen's (1989, 2007) stochastic complexity. It is rooted in the construction of complexity penalties for model selection (see procedure for VAR(n) model in Section 2.4), where a good trade-off between the prediction accuracy gained by increasing the number of free parameters and the danger of overfitting the model to random fluctuations and not regularities that generalize to other datasets has to be found. In an early paper,

Wallace and Boulton (1968) hypothesized that this trade-off could best be achieved by selecting the model with "the briefest recording of all attribute information." Akaike (1973, 1974) developed an important quantitative step along this line of thought by formulating a simple relationship between the expected Kullback-Leibler information and Fisher's maximized log-likelihood function (see deLeeuw 1992). He created his model selection criterion—which is today known as the Akaike Information Criterion (AIC, see Section 2.4)—without explicit links to complexity theory. Yet even from a complexity-theoretical perspective the AIC is not arbitrary, as it represents the asymptotic bias correction term of the maximized log-likelihood from each approximating model to full reality and can therefore be interpreted as a "complexity penalty" for increasing the number of free parameters beyond a point that is justified by the data (Burnham and Anderson 2002). Mathematically speaking, the AIC is defined as (Burnham and Anderson 2002)

$$AIC = -2 \ln \mathcal{L}\left(\widehat{\theta}|x\right) + 2k, \tag{201}$$

where the expression $\ln \mathcal{L}\left(\widehat{\theta}|x\right)$ denotes the numerical value of the log-likelihood at its maximum point, and k denotes the effective number of parameters (see Section 2.4). The maximum point of the log-likelihood function corresponds to the values of the maximum likelihood estimates $\widehat{\theta}$ of the free parameters of the approximating model given data x. In terms of a heuristic complexity-theoretic interpretation, the first term in AIC, $-2 \ln \mathcal{L}\left(\widehat{\theta}|x\right)$ can be considered as a measure of lack of model fit, while the second term $2k$ represents the cited complexity penalty for increasing the freely estimated parameters beyond a point that is compatible with the data-generating mechanisms. In the above definition, the dependency of the criterion on the number of data points is only implicit through the likelihood function. According to Section 2.4, for $VAR(n)$ models assuming normally distributed errors with a constant covariance, the dependency can easily be made explicit from least square regression statistics (Eq. 67) as

$$AIC(n) = \ln \mathrm{Det}\left[\widehat{\Sigma}_{(n)}\right] + \frac{2}{T}k,$$

where

$$\widehat{\Sigma}_{(n)} = \frac{\widehat{\Delta}_{(n)}}{T} \tag{202}$$

is the maximum likelihood estimate of the one-step prediction error of order n and

$$k = n\,p^2 + \frac{p(p+1)}{2}$$

denotes according to Eq. 68 the effective number of parameters related to the coefficient matrices $A_0, \ldots, A_{n-1}$ and the covariance matrix C of the inherent one-step prediction error (sensu Akaike 1973).

Akaike's fundamental ideas were systematically developed by Rissanen in a series of papers and books starting from 1978. Rissanen (1989, 2007) emphasizes that fitting a statistical model to data is equivalent to finding an efficient encoding of that data, and that in searching for an efficient code we need to measure not only the number of bits required to describe the deviations of the data from the model's predictions, but also the number of bits required to specify the independent parameters of the model (Bialek et al. 2001). This specification has to be made with a level of precision that is supported by the data.

To clarify this theoretically convincing concept, it is assumed that we carried out a work sampling study in a complex PD project involving many and intensive cooperative relationships between the development teams. Based on a large number of observations the proportion of time spent by the developers in predefined categories of activity $\mathcal{X} = \{x_1, \ldots, x_m\}$ (e.g. sketching, drawing, calculating, communicating etc.) was estimated with high statistical accuracy. In addition to the observations made at random times, a comprehensive longitudinal observation of the workflows of different development teams was carried out in a specific project phase at regular intervals. The observations were made in R independent trials and encoded by the same categories of activity $\mathcal{X}$. We define the r-th workflow in the specific project phase in formal terms as a data string $x_r^{\mathcal{T}} = \left(x_{j_r(0)}, \ldots, x_{j_r(\mathcal{T})}\right)$ of length $(\mathcal{T}+1)$ $\left(x_{j_r(\tau)} \in \mathcal{X},\ j_r(\tau) \in \{1, \ldots, |\mathcal{X}|\},\ \tau = 0, 1, \ldots, \mathcal{T}, r = 1, \ldots, R\right)$. In a similar manner as in the previous section the index $j_r(\tau)$ can be interpreted as a pointer to activity $x_{j_r(\tau)} \in \mathcal{X}$ observed at time instant τ in the r-th workflow encoded by $x_r^{\mathcal{T}}$. All empirically acquired data strings are stored in a database of ordered sequences $DB = \left\{x_1^{\mathcal{T}}, \ldots, x_R^{\mathcal{T}}\right\}$. We aim at developing an integrative workflow model that can be used for the prediction and evaluation of development activities in the project phase based on the theory of discrete random processes. Therefore, we start by defining a finite one-dimensional random process $(X_0, \ldots, X_{\mathcal{T}})$ of discrete state variables. In terms of information theory the process communicates to an observer how the development activities unfold and are organized in time. In formal terms, $(X_0, \ldots, X_{\mathcal{T}})$ is a joint ensemble $\mathbb{E}$, in which each outcome is an ordered sequence $\left(x_{j(0)}, \ldots, x_{j(\mathcal{T})}\right)$ with $x_{j(0)} \in \mathcal{X} = \{x_1, \ldots, x_m\}$, $x_{j(1)} \in \mathcal{X}$, $\ldots$, $x_{j(\tau)} \in \mathcal{X}$, $\ldots, x_{j(\mathcal{T})} \in \mathcal{X}$ (see, e.g. MacKay 2003). Each component X_τ of the joint ensemble $\mathbb{E} = X_0, \ldots, X_\tau, \ldots, X_{\mathcal{T}}$ is an ensemble. An ensemble X_τ is a triple $\left(x_{j(\tau)}, A_{X_\tau}, P_{X_\tau}\right)$, where the outcome $x_{j(\tau)}$ is the value of a random variable that can take on one of a set of possible values $A_X = \left(a_1, a_2, \ldots, a_{|\mathcal{X}|}\right)$, having probabilities $P_X = \left(p_1, p_2, \ldots, a_{|\mathcal{X}|}\right)$, with $P(X_\tau = a_i) = p_i$ (MacKay 2003). It holds that $p_i \geq 0$ and $\Sigma_{a_i \in A_X} P(X = a_i) = 1$. We call the term

$$P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right) \quad j(\tau) \in \{1, \ldots, |\mathcal{X}|\}$$

the joint probability of $\left(x_{j(0)}, \ldots, x_{j(\mathcal{T})}\right)$. The joint probability describes the statistical properties of the joint ensemble in the sense that, when evaluated at a given data point $\left(x_{j(o)}, \ldots, x_{j(\mathcal{T})}\right)$, we get the probability that the realization of the random

sequence will be equal to that data point. A joint ensemble is therefore a probability distribution on $\{x_1, \ldots, x_m\}^{\mathcal{T}}$. Similar to the definition of the probability density function of a continuous-type random variable from Section 2.2, we can make the functional relationship between the values and their joint probability explicit and use a joint probability mass function $p_{(X_0,\ldots,X_{\mathcal{T}})}$:

$$p_{(X_0,\ldots,X_{\mathcal{T}})}\left(x_{j(0)}, \ldots, x_{j(\mathcal{T})}\right) = P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right) \quad j(\tau) \in \{1, \ldots, |\mathcal{X}|\}.$$

The joint probability mass function completely characterizes the probability distribution of a joint ensemble. Without limiting the generality of the approach, the joint probability $P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right)$ of $\left(x_{j(o)}, \ldots, x_{j(\mathcal{T})}\right)$ as an integrative workflow model of the specific phase of the PD project can be factorized over all $\mathcal{T}$ time steps using, iteratively, the definition for the conditional probability $P(X|Y) = P(X,Y)/P(Y)$ as:

$$\begin{aligned} &P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right) \\ &= P\left(X_0 = x_{j(0)}\right) \prod_{\tau=1}^{\mathcal{T}} P\left(X_\tau = x_{j(\tau)} \middle| X_{\tau-1} = x_{j(\tau-1)}, \ldots, X_0 = x_{j(0)}\right). \end{aligned}$$

The above decomposition of the joint probability into conditional distributions $P\left(X_\tau = x_{j(\tau)} \middle| X_{\tau-1} = x_{j(\tau-1)}, \ldots, X_0 = x_{j(0)}\right)$ with correlations of increasing length τ can theoretically capture interactions between activities of long range and therefore holds true under any circumstances of cooperative relationships in the given phase. It is assumed that there are persistent workflow patterns in the project phase and we can express them by means of a reduced dependency structure capturing only short correlations, e.g. by using a Markov chain of order $n \ll \mathcal{T}$ or an equivalent dynamic Bayesian network (see Gharahmani 2001). As such, the reduced dependency structure reflects only the essential signature of spatiotemporal coordination in the project phase on a specific time scale. In the simplest case, only transitions between two consecutive development activities must be taken into account and a Markov chain of first order is an adequate candidate model for capturing these direct dynamic dependencies. In this model the conditional probability distribution of development activities at the next time step—and in fact all future steps—depends only on the current activity and not on past instances of the process when the current activity is known. Accordingly, the current activity shields the future from past histories, and the joint probability can be expressed as:

$$P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right) = P\left(X_0 = x_{j(0)}\right) \prod_{\tau=1}^{\mathcal{T}} P\left(X_\tau = x_{j(\tau)} \middle| X_{\tau-1} = x_{j(\tau-1)}\right).$$

After the model structure of the Markov chain of first order has been defined by the above factorization of the joint probability, we have to specify the free parameters.

Continuing the notation of the previous chapters we denote the parameter vector by $\theta \in \mathbb{R}^k$. Due to the intrinsic "memorylessness" of the chain, only the initial distribution

$$\pi_0 = \begin{pmatrix} P(X_0 = x_1) & \dots & P\big(X_0 = x_{|\mathcal{X}|}\big) \end{pmatrix} \in [0;1]^{|\mathcal{X}|}$$

of the probability mass over the state space $\mathcal{X}$ and the transition probabilities

$$P = (p_{ij}) = \begin{pmatrix} P\big(X_\tau = x_1 \big| X_{\tau-1} = x_1\big) & P\big(X_\tau = x_2 \big| X_{\tau-1} = x_1\big) & \dots \\ P\big(X_\tau = x_1 \big| X_{\tau-1} = x_2\big) & P\big(X_\tau = x_2 \big| X_{\tau-1} = x_2\big) & \\ \vdots & & \ddots \end{pmatrix} \in [0;1]^{|\mathcal{X}|^2}$$

between consecutive activities are relevant for making good predictions. Hence, we have the ordered pair of parameters:

$$\theta_1 = [\, \pi_0 \quad P \,] .$$

Note that only $(|\mathcal{X}| - 1)$ parameters of the initial distribution π_0 and $|\mathcal{X}|(|\mathcal{X}| - 1)$ of the transition matrix P are freely estimated parameters, because a legitimate probability distribution has to be formed and the constraints

$$\sum_{i=1}^{|\mathcal{X}|} \pi_0^{(i)} = 1 \quad \text{and} \quad \forall i : \sum_{j=1}^{|\mathcal{X}|} p_{ij} = 1$$

have to be satisfied.

We can use Maximum Likelihood Estimation (MLE, see Section 2.4) to minimize the deviations of the empirically acquired data sequences from the model's predictions (see e.g. Papoulis and Pillai 2002; Shalizi 2006). In other words, the goodness of fit is maximized. The maximum likelihood estimate of the parameter pair θ_1 is denoted by $\widehat{\theta}_{1,\mathcal{T}}$. MLE was pioneered by R. A. Fisher (cf. Edwards 1972) under a repeated-sampling paradigm and is the most prominent estimation technique. As an estimation principle, maximum likelihood is supported by $\widehat{\theta}_{1,\mathcal{T}}$'s asymptotic efficiency in a repeated sampling setting under mild regularity conditions and its attainment of the Cramér-Rao lower bound in many exponential family examples in the finite-sample case (Hansen and Yu 2001). For a first-order Markov chain, the estimate $\widehat{\theta}_{1,\mathcal{T}}$ can be determined by solving the objective function:

$$\begin{aligned} \widehat{\theta}_{1,\mathcal{T}} &= \arg\max_{\theta_1} \prod_{r=1}^{R} P\big(X_0 = x_{j_r(0)} | \theta_1\big) \prod_{\tau=1}^{\mathcal{T}} P\big(X_\tau = x_{j_r(\tau)} \big| X_{\tau-1} = x_{j_r(\tau-1)}, \theta_1\big) \\ &= \arg\max_{(\pi_0, P)} \prod_{r=1}^{R} P\big(X_0 = x_{j_r(0)} | \pi_0\big) \prod_{\tau=1}^{\mathcal{T}} P\big(X_\tau = x_{j_r(\tau)} \big| X_{\tau-1} = x_{j_r(\tau-1)}, P\big). \end{aligned}$$

Note that the objective function is only valid if all R data sequences had been acquired in independent trials.

Due to the inherent memorylessness of the first-order Markov chain, this model is usually not expressive enough to capture the complicated dynamic dependencies between activities in a project phase. Consequently, a second-order Markov chain is considered as a second approximating model with extended memory capacity. For this model, the joint probability can be expressed as:

$$P\big(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\big) = P\big(X_0 = x_{j(0)}\big) P\big(X_1 = x_{j(1)} \big| X_0 = x_{j(0)}\big) \cdot \prod_{\tau=2}^{\mathcal{T}} P\big(X_\tau = x_{j(\tau)} \big| X_{\tau-1} = x_{j(\tau-1)}, X_{\tau-2} = x_{j(\tau-2)}\big).$$

It is evident that the conditional distribution $P\big(X_\tau = x_{j(\tau)} \big| X_{\tau-1} = x_{j(\tau-1)}, X_{\tau-2} = x_{j(\tau-2)}\big)$ cannot only be used to predict direct transitions between current and future activities but can also model transitions between activities of the process that are conditioned on two time steps in the past. To parameterize this extended chain, three quantities are required: The initial distribution

$$\pi_0 = \begin{pmatrix} P(X_0 = x_1) & \ldots & P\big(X_0 = x_{|\mathcal{X}|}\big) \end{pmatrix} \in [0;1]^{|\mathcal{X}|},$$

the transition probabilities between consecutive activities at the first two time steps

$$P_0 = (p_{0,ij}) = \begin{pmatrix} P\big(X_1 = x_1 \big| X_0 = x_1\big) & P\big(X_1 = x_2 \big| X_0 = x_1\big) & \ldots \\ P\big(X_1 = x_1 \big| X_0 = x_2\big) & P\big(X_1 = x_2 \big| X_0 = x_2\big) & \\ \vdots & & \ddots \end{pmatrix} \in [0;1]^{|\mathcal{X}|^2}$$

and the transition probabilities for the next activity given both preceding activities at arbitrary time steps

$$P = (p_{ij}) = \begin{pmatrix} p\big(x_1 \big| x_1, x_1\big) & p\big(x_1 \big| x_1, x_2\big) & \cdots & p\big(x_1 \big| x_1, x_{|\mathcal{X}|}\big) \\ p\big(x_1 \big| x_2, x_1\big) & p\big(x_1 \big| x_2, x_2\big) & \cdots & p\big(x_1 \big| x_2, x_{|\mathcal{X}|}\big) \\ \vdots & \vdots & & \vdots \\ p\big(x_1 \big| x_{|\mathcal{X}|}, x_1\big) & p\big(x_1 \big| x_{|\mathcal{X}|}, x_2\big) & & p\big(x_1 \big| x_{|\mathcal{X}|}, x_{|\mathcal{X}|}\big) \\ p\big(x_2 \big| x_1, x_1\big) & p\big(x_2 \big| x_1, x_2\big) & & p\big(x_2 \big| x_1, x_{|\mathcal{X}|}\big) \\ p\big(x_2 \big| x_2, x_1\big) & p\big(x_2 \big| x_2, x_2\big) & & p\big(x_2 \big| x_2, x_{|\mathcal{X}|}\big) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ p\big(x_{|\mathcal{X}|} \big| x_{|\mathcal{X}|}, x_1\big) & p\big(x_{|\mathcal{X}|} \big| x_{|\mathcal{X}|}, x_2\big) & & p\big(x_{|\mathcal{X}|} \big| x_{|\mathcal{X}|}, x_{|\mathcal{X}|}\big) \end{pmatrix} \in [0;1]^{|\mathcal{X}|^3}.$$

In the above matrix the shorthand notation $p\big(x_i \big| x_j, x_k\big) = P\big(X_\tau = x_i \big| X_{\tau-1} = x_j, X_{\tau-2} = x_k\big)$ was used. Hence, we have the parameter triple

$$\theta_2 = [\, \pi_0 \quad P_0 \quad P \,].$$

In this triple $(|\mathcal{X}| - 1)$ parameters of the initial distribution π_0, $|\mathcal{X}|(|\mathcal{X}| - 1)$, parameters of the initial transition matrix P_0 and $|\mathcal{X}|^2(|\mathcal{X}| - 1)$ of the general transition matrix P are freely estimated parameters, because a legitimate probability

distribution has to be formed. The ordered pair $[\pi_0 \quad P_0]$ can be regarded as the initial state of the chain. We denote the maximum likelihood estimate for the parameterized model by $\widehat{\theta}_{2,\mathcal{T}}$. The corresponding objective function is:

$$\begin{aligned}\widehat{\theta}_{2,\mathcal{T}} &= \arg\max_{\theta_2} \prod_{r=1}^{R} P\left(X_0 = x_{j_r(0)} | \theta_2\right) P\left(X_1 = x_{j_r(1)} \middle| X_0 = x_{j_r(0)}, \theta_2\right) \\ &\quad \cdot \prod_{\tau=2}^{\mathcal{T}} P\left(X_\tau = x_{j_r(\tau)} \middle| X_{\tau-1} = x_{j_r(\tau-1)}, X_{\tau-2} = x_{j_r(\tau-2)}, \theta_2\right) \\ &= \arg\max_{(\pi_0, P_0, P)} \prod_{r=1}^{R} P\left(X_0 = x_{j_r(0)} | \pi_0\right) P\left(X_1 = x_{j_r(1)} \middle| X_0 = x_{j_r(0)}, P_0\right) \\ &\quad \cdot \prod_{\tau=2}^{\mathcal{T}} P\left(X_\tau = x_{j_r(\tau)} \middle| X_{\tau-1} = x_{j_r(\tau-1)}, X_{\tau-2} = x_{j_r(\tau-2)}, P\right).\end{aligned}$$

It is not difficult to prove that the solutions of the objective functions for Markov chains of first and second order (as well as all higher orders) are equivalent to the relative frequencies of observed subsequences of activity in the database *DB* (Papoulis and Pillai 2002). In other words, the MLE results can be obtained by simple frequency counting of data substrings of interest. Let the #-operator be a unary counting operator that counts the number of times the data string $(x_{j(o)} x_{j(1)} \ldots)$ in the argument occurred in $DB = \left\{x_1^{\mathcal{T}}, \ldots, x_R^{\mathcal{T}}\right\}$. Then the MLE yields

$$\widehat{\pi}_0 = \left\{\frac{1}{R}\right\} \cdot \begin{pmatrix} \#(x_1)_{\tau=0} & \ldots & \#\left(x_{|\mathcal{X}|}\right)_{\tau=0} \end{pmatrix}$$

$$\widehat{P} = \left\{\frac{1}{R\mathcal{T}}\right\} \cdot \begin{pmatrix} \#(x_1x_1) & \#(x_1x_2) & \ldots \\ \#(x_2x_1) & \#(x_2x_2) & \\ \vdots & & \ddots \end{pmatrix}$$

for the first-order Markov chain and

$$\widehat{\pi}_0 = \left\{\frac{1}{R}\right\} \cdot \begin{pmatrix} \#(x_1)_{\tau=0} & \ldots & \#\left(x_{|\mathcal{X}|}\right)_{\tau=0} \end{pmatrix}$$

$$\widehat{P}_0 = \left\{\frac{1}{R}\right\} \cdot \begin{pmatrix} \#(x_1x_1)_{\tau=0} & \#(x_1x_2)_{\tau=0} & \ldots \\ \#(x_2x_1)_{\tau=0} & \#(x_2x_2)_{\tau=0} & \\ \vdots & & \ddots \end{pmatrix}$$

$$\widehat{P} = \left\{\frac{1}{R(\mathcal{T}-1)}\right\} \cdot \begin{pmatrix} \#(x_1x_1x_1) & \#(x_2x_1x_1) & \cdots & \#\left(x_{|\mathcal{X}|}x_1x_1\right) \\ \#(x_1x_2x_1) & \#(x_2x_2x_1) & \cdots & \#\left(x_{|\mathcal{X}|}x_2x_1\right) \\ \vdots & \vdots & & \vdots \\ \#\left(x_1x_{|\mathcal{X}|}x_1\right) & \#\left(x_2x_{|\mathcal{X}|}x_1\right) & & \#\left(x_{|\mathcal{X}|}x_{|\mathcal{X}|}x_1\right) \\ \#(x_1x_1x_2) & \#(x_2x_1x_2) & & \#\left(x_{|\mathcal{X}|}x_1x_2\right) \\ \#(x_1x_2x_2) & \#(x_2x_2x_2) & & \#\left(x_{|\mathcal{X}|}x_2x_2\right) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \#\left(x_1x_{|\mathcal{X}|}x_{|\mathcal{X}|}\right) & \#\left(x_2x_{|\mathcal{X}|}x_{|\mathcal{X}|}\right) & & \#\left(x_{|\mathcal{X}|}x_{|\mathcal{X}|}x_{|\mathcal{X}|}\right) \end{pmatrix}$$

for the second-order chain. To estimate the initial state probabilities $\widehat{\pi}_0$ only the observations $\left(\#(x_1)_{\tau=0} \quad \ldots \quad \#\left(x_{|\mathcal{X}|}\right)_{\tau=0}\right)$ in the first time step $\tau = 0$ must be counted. To calculate the initial transition matrix P_0 of the Markov chain of second order, only the data points in the first two time steps have to be considered, and we therefore use $\#(x_\cdot x_\cdot)_{\tau=0}$ to indicate the number of all leading substrings of length two. The estimate of the initial state distribution can be refined by using the data from the cited work sampling study that was carried out prior to the longitudinal observation of workflows.

The above solutions show that in a complex PD project that already manifests its intrinsic complexity in a single project phase by a rich body of data sequences with higher-order correlations, the data can usually be predicted much better with a second-order Markov chain than with a first-order model. This is due to the simple fact that the second-order chain has additional $|\mathcal{X}|^2(|\mathcal{X}| - 1)$ free parameters for encoding specific activity patterns and therefore a larger memory capacity. By inductive reasoning we can proceed with nesting Markov models of increasing order n

$$\begin{aligned} &P\left(X_0 = x_{j(0)}, \ldots, X_{\mathcal{T}} = x_{j(\mathcal{T})}\right) = \\ &P\left(X_0 = x_{j(0)}\right) P\left(X_1 = x_{j(1)} \middle| X_0 = x_{j(0)}\right) \ldots \\ &P\left(X_{n-1} = x_{j(n-1)} \middle| X_{n-2} = x_{j(n-2)}, \ldots, X_0 = x_{j(0)}\right) \\ &\cdot \prod_{\tau=n}^{\mathcal{T}} P\left(X_\tau = x_{j(\tau)} \middle| X_{\tau-1} = x_{j(\tau-1)}, \ldots, X_{\tau-n} = x_{j(\tau-n)}\right) \end{aligned} \tag{203}$$

and capture more and more details of the workflows. Formally speaking, the n-th order Markov model is the set of all n-th order Markov chains, i.e. all statistical representations that are equipped with a starting state and satisfy the above factorization of the joint probability. Given the order n of the chain, the probability distribution of X_τ depends only on the n observations preceding τ. However, beyond an order that is supported by the data, we begin to encounter the problem of "not seeing the forest for the trees" and incrementally fitting the model to random fluctuations that do not generalize to other datasets from the same project phase.

In order to avoid this kind of overfitting, the maximum likelihood paradigm has to be extended, because for an approximating model of interest, the likelihood function only reflects the conformity of the model to the data. As the complexity of the model is increased and more freely estimated parameters are included, the model usually becomes more capable of adapting to specific characteristics of the data. Therefore, selecting the parameterized model that maximizes the likelihood often leads to choosing the most complex model in the approximating set. Rissanen's minimum description length (MDL) principle (1989) provides a natural safeguard against overfitting by using the briefest encoding of not only the attribute information related to the data sequences but also to the parameters of the approximating models. In general, let θ be a parameter vector of model

$$\mathcal{M}^{(n)} = \left\{P\left(X_0 = x_{j(0)}, \ldots, X_\mathcal{T} = x_{j(\mathcal{T})}\middle|\theta\right) : \theta \in \Theta \subset \mathbb{X}^n\right\}$$

whose support is a set $\mathbb{X}$ of adequate dimensionality and consider the class

$$\mathcal{M} = \bigcup_{n=1}^{N} \mathcal{M}^{(n)}$$

consisting of all models represented by parametric probability distributions $P\left(X_0 = x_{j(0)}, \ldots, X_\mathcal{T} = x_{j(\mathcal{T})}\middle|\theta\right)$ from the first order up to order N (Rissanen 2012). Note that Rissanen (2012) also calls $\mathcal{M}^{(n)}$ a model class that is defined by the independent parameters. For the sake of simplicity and to remain consistent with the previously used notation, we simply speak of an approximating model. The sequence of discrete state variables $\left(X_0, \ldots, X_\mathcal{T}\middle|\theta\right)$ forms a one-dimensional random process encoding a joint ensemble of histories that can be explained by the structure and independent parameters of an approximating model within the class $\mathcal{M}$. By using a model with a specific structure and parameters, the joint probability can usually be decomposed into predictive distributions whose conditional part does not scale with the length of the sequence and therefore does not need an exponentially growing number of freely estimated parameters. In the following the number of parameters incorporated in the vector θ is the only variable of interest that is related to a specific model representation within class $\mathcal{M}$.

As previously shown, a model from class $\mathcal{M}$ with parameter vector θ assigns a certain probability

$$p_\theta\left(x^\mathcal{T}\right) = P\left(X_0 = x_{j(0)}, \ldots, X_\mathcal{T} = x_{j(\mathcal{T})}\middle|\theta\right) \tag{204}$$

to a data sequence $\left(x_{j(0)}, \ldots, x_{j(\mathcal{T})}\right)$ of interest. If we take the definition of the Shannon information content of an ensemble X

$$I[x] := \log_2 \frac{1}{P(X = x)}, \tag{205}$$

then the likelihood function $p_\theta\left(x^\mathcal{T}\right)$ can be transformed into an information-theory loss function L

$$\begin{aligned} L\left[\theta, x^\mathcal{T}\right] &= I\left[p_\theta\left(x^\mathcal{T}\right)\right] \\ &= \log_2 \frac{1}{P\left(X_0 = x_{j(0)}, \ldots, X_\mathcal{T} = x_{j(\mathcal{T})}\middle|\theta\right)} \\ &= -\log_2 P\left(X_0 = x_{j(0)}, \ldots, X_\mathcal{T} = x_{j(\mathcal{T})}\middle|\theta\right). \end{aligned} \tag{206}$$

According to Eq. 203 we can interpret $L\left[\theta, x^\mathcal{T}\right]$ in a predictive view as the loss incurred when forecasting X_τ sequentially based on the conditional distributions $P\left(X_\tau = x_{j(\tau)}\middle|X_{\tau-1} = x_{j(\tau-1)}, \ldots, X_{\tau-n} = x_{j(\tau-n)}, \theta\right)$. The loss is measured using a logarithmic scale. In the predictive view MLE aims at minimizing the accumulated

logarithmic loss. We denote the maximum likelihood estimate by the member $\widehat{\theta}_T$. In the sense of information theory, minimizing the loss can also be thought of as minimizing the encoded length of the data based on an adequate prefix code $d(x)$. Shannon's famous source coding theorem (see e.g. Cover and Thomas 1991) tells us that for an ensemble X there exists a prefix code $d(x)$ with expected length $L[d(x), X]$ satisfying

$$-\sum_{x\epsilon\mathcal{X}} P(X=x)\log_2 P(X=x) \leq L[d(x),X] < -\sum_{x\epsilon\mathcal{X}} P(X=x)\log_2 P(X=x)+1. \quad (207)$$

The term on the left of the inequality is the "information entropy" (see Eq. 210). It measures in [bits] the amount of freedom of choice in the coding process. This fundamental quantity will be explained in detail in the next chapter. A beautifully simple algorithm for finding a prefix code with minimal expected length is the Huffman coding algorithm (see e.g. Cover and Thomas 1991). In this algorithm the two least probable data points in $\mathcal{X}$ are taken and assigned the longest codewords. The longest codewords are of equal length and differ only in the last digit. In the next step, these two symbols are combined into a new single symbol and the procedure is repeated. Since each recursion reduces the size of the alphabet by one, the algorithm will have assigned strings to all symbols after $|\mathcal{X}|-1$ steps. Following the predictive view, we can obtain an intuitive interpretation of the logarithmic loss in terms of coding: the code length needed to encode the data points $\left(x_{j(0)},\ldots,x_{j(\mathcal{T})}\right)$ with prefix code $d(x)$ based on the joint distribution $P(.)$ is simply the accumulated logarithmic loss incurred when the corresponding conditional distributions $P(.\,|.)$ are used to sequentially predict the τ-th outcome on the basis of the previous $(\tau-1)$ observations (Grünwald 2007).

It is evident that this interpretation is incomplete; we have an encoded version of the data, but we have still not said what the encoding scheme for the member $\widehat{\theta}_T$ is. Thus, the total description length DL must be divided into two parts,

$$DL\left[x^{\mathcal{T}},\theta,\Theta\right] = L\left[\theta,x^{\mathcal{T}}\right] + D[\theta,\Theta],$$

where $D[\theta,\Theta]$ denotes the code length in terms of the number of bits needed to specify the member within class $\mathcal{M}$. The two parts of description length are usually obtained in a sequential two-stage encoding process (see Hansen and Yu 2001). In the first stage, the description length $D\left[\widehat{\theta}_T,\Theta\right]$ for the best-fitting member $\widehat{\theta}_T$ is calculated. The $\widehat{\theta}_T$'s maximizing the goodness-of-fit can be obtained both by MLE and Bayesian estimation. In the second stage, the description length of data $L\left[\widehat{\theta}_T,x^{\mathcal{T}}\right]$ is determined on the basis of the parameterized probability mass function $p_{\hat{\theta}_T}\left(x^{\mathcal{T}}\right)$.

Clearly, the model related to $D[\theta, \Theta]$ represents the part of the description that can be generalized, while $L\left[\theta, x^{\mathcal{T}}\right]$ includes the noisy part that does not generalize to other datasets. If $D[\theta, \Theta]$ assigns short code words to simple models, we have the desired tradeoff: we can reduce the part of the data that looks like noise only by using a more elaborate approximating model. Such an assignment provides an effective safeguard against overfitting. The minimum description length (MDL) principle supplied by Rissanen (1989, 2007) allow us to select the model that minimizes the total description length:

$$\theta_{MDL} := \arg\min_{\theta} DL\left[x^{\mathcal{T}}, \theta, \Theta\right].$$

The only requirement for the code length of the optimizing parameters $D\left[\widehat{\theta}_{\mathcal{T}}, \Theta\right]$ of this general MDL principle is that they be decodable (Rissanen 2012). The definition of a prior probability as in Bayesian estimation is therefore not required. Minimizing the total description length is apparently a consistent principle in connection with maximum likelihood estimation, because if we want to maximize the joint probability $DL\left[x^{\mathcal{T}}, \theta, \Theta\right]$ we need to calculate the probability of the coincidence of the observed data and the different approximating models and choose the maximizing model. It is important to point out that in MDL, one is never concerned with actual encodings but only with code length functions, e.g. $L[d(x), X]$ for an ensemble X encoded by a prefix code $d(x)$ (Grünwald 2007). The stochastic complexity C_{SC} of the joint ensemble $X^{\mathcal{T}}$ with reference to the model class $\mathcal{M}$ is simply the MDL:

$$C_{SC}\left[x^{\mathcal{T}}, \Theta\right] := \min_{\theta} DL\left[x^{\mathcal{T}}, \theta, \Theta\right]. \tag{208}$$

Under mild conditions for the underlying data-generating process in the model class, as we provide more data, θ_{MDL} will converge to the model that minimizes the generalization error.

Returning to our previous example of workflow modeling with Markov chains, we can follow the considerations of Hansen and Yu (2001) and, for didactic purposes, construct a simple but reasonable two-part code for the n-th order Markov chain $\mathcal{M}^{(n)}$ within the class $\mathcal{M}$ of finite-order Markov chains up to order N. The parameter vector of the n-th order Markov chain is denoted by $\theta_n \in \Theta_n$. Firstly, the order has to be described. We can start with a straightforward, explicit description for n that is based on a binary prefix code with $\lceil \log_2 n \rceil$ zeros followed by a one. The encoding of n can be done by using a simple uniform code for $\left\{1, \ldots, 2^{\lceil \log_2 n \rceil}\right\}$. Therefore, we need approximately $2\lceil \log_2 n \rceil + 1$ bits to describe the model order. By applying Huffman's algorithm here, we can also obtain a more efficient uniform code with a length function that is not greater than $\lfloor \log_2 n \rfloor$ for all values of $\{1, 2, \ldots, n\}$ but is equal to $\lfloor \log_2 n \rfloor$ for at least two values in this set. The function $\lfloor . \rfloor$ provides the integer part of the argument. Whereas we know from Shannon's source coding theorem (Eq. 207) that an expected length of such a code is optimal

only for a true uniform distribution of the order of the model, this code is a reasonable choice when little is known about how the data was generated. Secondly, the $\Sigma_{i=0}^{n}|\mathcal{X}|^{i}(|\mathcal{X}|-1)=|\mathcal{X}|^{n+1}$ best-fitting free parameters $\widehat{\theta}_{n,\mathcal{T}}$ have to be described. We start by discretizing the range $[0;1]$ of a single ensemble into equal cells of size δ and then apply Huffman's algorithm. If we discretize the Cartesian product $\Theta_n=[0;1]^{|\mathcal{X}|^{n+1}}$ associated with the joint ensemble $X^{\mathcal{T}}$ in the same fashion, the quantity $-\log_2\left(p\left([0;1]^{|\mathcal{X}|^{n+1}}\right)\cdot\delta^{|\mathcal{X}|^{n+1}}\right)=-\log_2 p\left([0;1]^{|\mathcal{X}|^{n+1}}\right)-|\mathcal{X}|^{n+1}\log_2\delta$ can be viewed as the code length of a prefix code for $\widehat{\theta}_{n,\mathcal{T}}$ (Hansen and Yu 2001). Here, the probability density p can be regarded as an auxiliary density. It is used instead of the unknown true parameter-generating density f. Assuming a continuous uniform distribution with density $p(x)=1$ for $x\in[0;1]^{|\mathcal{X}|^{n+1}}$ (and $q(x)=0$ otherwise), an additional $|\mathcal{X}|^{n+1}\log_2\delta$ bits are needed to describe the free parameters. In a compact parameter space, we can refine the description and choose for the precision $\delta=\sqrt{1/(\mathcal{T}+1)}$ for each effective dimension. Rissanen (1989) showed that this choice of precision is optimal in regular parametric families. The intuitive explanation is that $\sqrt{1/(\mathcal{T}+1)}$ represents the magnitude of the estimation error in $\widehat{\theta}_{n,\mathcal{T}}$ and therefore there is no need to encode the estimator with greater precision (Hansen and Yu 2001). When the uniform encoder is used, one needs a total of $\left(|\mathcal{X}|^{n+1}/2\right)\log_2(\mathcal{T}+1)$ bits to communicate an estimated parameter $\widehat{\theta}_{n,\mathcal{T}}$ of dimension $|\mathcal{X}|^{n+1}$. Putting both partial descriptions together leads to

$$D[\theta_n,\Theta_n]=\log_2 n+\frac{|\mathcal{X}|^{n+1}}{2}\log_2(\mathcal{T}+1).$$

Interestingly, the formalized total description length of the n-th order Markov chain is similar to the Schwarz-Bayes Criterion (BIC) for the $VAR(n)$ (Eq. 71) and LDS (Eq. 189) models of cooperative work in the sense that model complexity is penalized with a factor that increases linearly in the number of free parameters and logarithmically in the number of observations in the joint ensemble. This is a clear and unambiguous indication that there are deep theoretical connections between different approaches to model selection. The predictive view of Markovian models provides us with a refined interpretation of model selection based on the MDL principle: given two approximating models $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$, we should prefer the model that minimizes the accumulated prediction error resulting from a sequential prediction of future outcomes given all past histories (Grünwald 2007).

Regarded as a principle of model selection, MDL has proven very successful in many areas of application (see e.g. Grünwald 2007; Rissanen 2007). Nevertheless, a part of this success comes from carefully tuning the model-coding term $D[\theta,\Theta]$ in such a manner that those models that do not generalize well turn out to have long encodings. Though not illegitimate, this approach relies on the intuition and knowledge of the human model builder. Motivated in part by this kind of theoretical

incompleteness, Rissanen (2012) refined the above general MDL principle in his latest textbook on optimal estimation of parameters, formulating a "complete MDL principle." The complete MDL principle differs from the previously formulated principle in the requirement that the code length for the parameters defining the model $\mathcal{M}^{(k)}$ is the negative logarithm of the probability defined by the joint distribution

$$\hat{p}_k\left(x^{\mathcal{T}}\right) = \frac{p_{\hat{\theta}\left(x^{\mathcal{T}}\right)}\left(x^{\mathcal{T}}\right)}{\hat{C}_k},$$

where $\hat{C}_k$ is a normalizing coefficient. $\widehat{\theta}\left(x^{\mathcal{T}}\right)$ represents the ML estimator and k denotes the number of parameters incorporated in the parameter vector θ (Rissanen 2012). The requirement for the code length can also be generalized to the case where even the number of parameters is estimated, see Rissanen (2012). Since $\hat{p}_k\left(x^{\mathcal{T}}\right)$ is determined by the model $\mathcal{M}^{(k)}$, its code length is common for all data sequences. The code of $\hat{p}_k\left(x^{\mathcal{T}}\right)$ for fixed k is complete. The logarithm of the normalizing coefficient is given by the maximum capacity for the model $\mathcal{M}^{(k)}$ within class $\mathcal{M}$:

$$\log_2 \hat{C}_k = \log_2 \int_{\Theta} \sum\nolimits_{x^{\mathcal{T}}:\hat{\theta}\left(x^{\mathcal{T}}\right)=\theta} p_{\hat{\theta}\left(x^{\mathcal{T}}\right)}\left(x^{\mathcal{T}}\right) d\theta > 0.$$

The range Θ of the integration is selected to make the integral finite. Rissanen (2012) also calls the term $\log_2 \hat{C}_k$, representing the maximum capacity for model $\mathcal{M}^{(k)}$, the maximum estimation information, and interprets it as a measure of the maximum amount of information an estimator can obtain about the corresponding distribution. The estimator maximizing the estimation information agrees with the standard ML estimator. The model related to $\hat{p}_k\left(x^{\mathcal{T}}\right)$ was introduced earlier by Shtarkov (1987) as a universal information-theoretic model for data compression.

In spite of these recent refinements, the complete MDL principle has limitations in terms of selecting an adequate family of model classes. An additional shortcoming is non-optimality if the model class cannot be well defined (Rissanen 2007, 2012). Whatever its merits as a model selection method, stochastic complexity is not a good metric of emergent complexity in open organizational systems for three reasons (sensu Shalizi 2006). (1) The dependence on the model-encoding scheme is very difficult to formulate in a valid form for project-based organizations. (2) The log-likelihood term, $L\left[\theta, x^{\mathcal{T}}\right]$, can be decomposed into additional parts, one of which is related to the entropy rate of the information-generating work processes (h_μ, Eq. 223) and which therefore reflects their intrinsic unpredictability, not their complexity. Other parts indicate the degree to which even the most accurate model in $\mathcal{M}$ is misspecified, for instance, through an improper choice of the coordinate system. Thus, it largely reflects our unconscious incompetence as modelers, rather than a fundamental characteristic of the process. (3) The stochastic complexity

reflects the need to specify some particular organizational model and to formally represent this specification. This is necessarily part of the process of model development but seems to have no significance from a theoretical point of view. For instance, a sociotechnical system being studied does not need to represent its organization; it simply has it (Shalizi 2006).

3.2.3 *Effective Complexity*

Effective complexity (EC) was developed by Seth Lloyd and the Nobel laureate Murray Gell-Mann. The fact that random strings without any purposeful informational structure display maximal Kolmogorov−Chaitin complexity (see Section 3.2.1) was one of the main reasons for Gell-Mann and Lloyd's criticism of the algorithmic complexity concept from Section 3.2.1 and for their attempt to define effective complexity as a more intuitive measure for scientific discourse. The concept of EC and its mathematical treatment were the subject of a series of papers that gained a great deal of attention in the scientific community (Gell-Mann 1995; Gell-Mann and Lloyd 1996, 2004). As with previous approaches for evaluating the complexity of an entity with inherent regularities in terms of its structure and behavior, it is assumed that its complexity is manifested to an observer in the form of a data string x, typically encoded in binary form. However, Gell-Mann and Lloyd do not consider the minimum description length of the string itself, which is what Wallace and Boulton (1968) and Rissanen (1989, 2007) did to evaluate stochastic complexity. Instead, they consider the joint ensemble $\mathbb{E}$ in which the string is embedded as a typical member (Ladyman et al. 2013). "Typicality" is defined using the theory of types (see e.g. Cover and Thomas 1991), which means that the negative binary logarithm of the joint probability distribution of $\mathbb{E}[x]$ on $\{x_1, \ldots, x_m\}^{\mathcal{T}}$ is approximately equal to the information entropy $H[\mathbb{E}]$ (see below and next chapter). To evaluate the minimum description length of the ensemble $\mathbb{E}$, the (prefix) Kolmogorov−Chaitin complexity from Eq. 200 is used. This approach assumes that one can find a meaningful way to estimate what the ensemble is. The resulting informal definition of the EC[x] of an entity is the Kolmogorov−Chaitin complexity of the ensemble $\mathbb{E}$, in which the string x manifesting the object's complexity to an observer is embedded as a δ-typical member. Instead of Kolmogorov−Chaitin complexity, Gell-Mann and Lloyd use the equivalent term "algorithmic information content" (Gell-Mann and Lloyd 1996, 2004). The main idea of EC is therefore to split the algorithmic information content of the string x into two parts, where the first contains all regularities and the second contains all random features. The EC of x is defined as the algorithmic information content of the regularities alone (Ay et al. 2010). In contrast to previous approaches, the EC is therefore not a metric for evaluating the difficulty of describing all the attribute information of an entity, but rather the degree of organization (Ladyman et al. 2013). By degree of organization, we mean the internal structural and

behavioral regularities that can be identified by using ensembles as models of the string. Following this concept of ensemble-based complexity measurement, in order to compute the ensemble $\mathbb{E}$ a computer program on a universal computer U takes as input the target string x and a precision parameter n and simply outputs $\mathbb{E}[x]$ to precision n. This approach can resolve the paradox from Section 3.2.1, whereby random strings without any internal structure display high Kolmogorov−Chaitin complexity because no underlying regularities or rules exist that could allow a shorter description. The ideal ensemble for modeling a random string is a joint ensemble with a uniform distribution of the probability mass that assigns equal probability to every string x' of length $|x|$, and it holds that (Foley and Oliver 2011):

$$\mathbb{E}_x^U[x'] = 2^{-|x|}.$$

The Kolmogorov−Chaitin complexity of this ensemble is apparently very low, because the computer program used to calculate it on U simply calculates $2^{-|x|}$ to precision n when confronted with input x'. The EC of a random string is thus low, although it is incompressible and the Kolmogorov−Chaitin complexity is maximal for its length $|x|$ (Foley and Oliver 2011).

Ay et al. (2010) introduced a more formal approach to defining EC and proving some of its basic properties. In the following, we summarize their main definitions and interpretations. First, we have to define the Kolmogorov−Chaitin complexity $K_U[\mathbb{E}]$ of a joint ensemble $\mathbb{E}$. As previously stated, a program to compute the ensemble $\mathbb{E}$ on a universal prefix computer U expects two inputs: the target string x and a precision parameter $n \epsilon \mathbb{N}$. It outputs the binary digits of the approximation $\mathbb{E}_x^U$ of $\mathbb{E}[x]$ with an accuracy of at least 2^{-n}. The Kolmogorov−Chaitin complexity $K_U[\mathbb{E}]$ of $\mathbb{E}$ is then the length of the shortest program for the universal prefix computer U that computes $\mathbb{E}$ on the basis of the approximation $\mathbb{E}_x^U$ (Ay et al. 2010). Unfortunately, not every ensemble is computable, as there is a continuum of string ensembles but only a finite number of algorithms computing ensembles. Another subtlety is that the information entropy $H[\mathbb{E}] = \Sigma_{x \epsilon \mathcal{X}^\tau} \mathbb{E}(x) \log_2 \mathbb{E}(x)$ (cf. Eq. 210) as a measure of the "ignorance" of the probability distribution of a computable ensemble $\mathbb{E}(x)$ for string x does not necessarily need to be computable. All that is known is that it can be enumerated from "below." Thus, it must be assumed in the following that all ensembles are computable and have computable and finite entropy. Even when we restrict the analysis to the set of ensembles that are computable and have computable and finite entropy, the map $\mathbb{E} \mapsto H[\mathbb{E}]$ is not necessarily a computable function. Hence, the approximate equality $K_U[\mathbb{E}, H[\mathbb{E}]] \pm K_U[\mathbb{E}]$ is not necessarily uniformly true in $\mathbb{E}$ (the operator $\pm$ denotes an equality to within a constant). Therefore, the definition of $K_U(\mathbb{E})$ has to be refined (Ay et al. 2010):

$$K_U[\mathbb{E}] := K_U[\mathbb{E}, H[\mathbb{E}]].$$

We therefore assume that the programs on the universal prefix computer U computing $\mathbb{E}$ when confronted with input x carry an additional subroutine

to compute the information entropy $H[\mathbb{E}]$. The Kolmogorov−Chaitin complexity $K_U[\mathbb{E}]$ is integer-valued. Second, we have to define the "total information" $\Sigma[\mathbb{E}]$ of an ensemble $\mathbb{E}$ (Gell-Mann and Lloyd 1996, 2004). To explain the role of the total information within the theory, Gell-Mann and Lloyd (2004) consider a typical situation in which a theoretical scientist is trying to construct a theory to explain a large body of data. The theory is represented by a probability distribution over a set of bodies of data. One body consists of the real data, while the rest of the bodies are imagined. In this setting, the Kolmogorov−Chaitin complexity $K_U[\mathbb{E}]$ corresponds to the complexity of the theory, and the information entropy $H[\mathbb{E}]$ measures the extent to which the predictions of the theory are distributed widely over different possible bodies of data. Ideally, the theorist would like both quantities to be small: the Kolmogorov−Chaitin complexity $K_U[\mathbb{E}]$ so as to make the theory simple, and the information entropy $H[\mathbb{E}]$ so as to make it focus narrowly on the real data points. However, there can be a trade-off. By adding more details to the theory and more arbitrary parameters, the theoretical scientist might be able to focus on the real data, but only at the expense of complicating the theory. Similarly, by allowing appreciable probabilities for many possible bodies of data, the scientist might be able to develop a simple theory. In any case, it makes good sense to minimize the sum of the two quantities that is defined as the total information $\Sigma[\mathbb{E}]$:

$$\Sigma[\mathbb{E}] := K_U[\mathbb{E}] + H[\mathbb{E}].$$

This allows the scientist to deal with the possible trade-off: a good estimate of the ensemble that generated the string x should not only have a small Kolmogorov−Chaitin complexity and therefore provide a simple explanation in the language of U; it should also have a small information entropy, as the explanation should have a low level of arbitrariness and prefer outcomes that include the string x. The total information is a real number larger than or equal to one. Third, we have to explain what is meant by an ensemble $\mathbb{E}$ in which the string is embedded as a typical member. As previously stated, typicality is defined according to the theory of types (see, e.g. Cover and Thomas 1991). To briefly explain the concept of typicality, suppose that we toss a biased coin with probability p that it lands on heads and $q = 1 - p$ that it lands on tails n times. We call the resulting probability distribution the ensemble $\mathbb{E}'$. It is well known from theoretical and empirical considerations that typical outcomes x have a probability $\mathbb{E}'[x]$ that is close to 2^{-nH} (Cover and Thomas 1991). In this case the information entropy is defined as $H := -p \log_2 p - q \log_2 q$. We can prove that the probability that $\mathbb{E}'[x]$ lies between $2^{-n(H+\varepsilon)}$ and $2^{-n(H-\varepsilon)}$ for $\varepsilon > 0$ tends to one as the number of tosses n grows. This property is a simple consequence of the weak law of large numbers and is the subject of the "asymptotic equipartition theorem" (Cover and Thomas 1991). Generalizing this property, we consider a string x as typical for a joint ensemble $\mathbb{E}$ if its probability is not much smaller than $2^{-nH[\mathbb{E}]}$. We say x is δ–typical for $\mathbb{E}$ for some small constant $\delta \geq 0$ if

$$\mathbb{E}[x] \geq 2^{-H[\mathbb{E}](1+\delta)}.$$

Fourth, we have to define how small the total information $\Sigma[\mathbb{E}]$ should be for an ensemble $\mathbb{E}$ that explains the string x well but is not unnecessarily complex in the language of U. This lemma by Ay et al. (2010) shows that the total information should not be too small: it uniformly holds for $x \epsilon \mathcal{X}^*$ and $\delta \geq 0$ that

$$\frac{K_U(x)}{1+\delta} <^+ \text{Inf}\{\Sigma[\mathbb{E}] : x \text{ is typical for } \mathbb{E}\} <^+ K_U(x).$$

The symbol $<^+$ denotes an inequality to within a constant. $K_U(x)$ is the (algorithmic) Kolmogorov−Chaitin complexity of x according to Eq. 200. The function Inf{.} denotes the infimum of the generated set of total information values. Put simply, the lemma tells us that the total information $\Sigma[\mathbb{E}]$ should not be much larger than the Kolmogorov−Chaitin complexity of the string of interest. Fifth, the ultimate question is, of all the "good" ensembles according to the previously defined criteria, which ensemble $\mathbb{E}$ is the best for evaluating an entity's degree of organization? In their simple yet convincing answer, Gell-Mann and Lloyd (1996, 2004) claim it is the ensemble with minimum Kolmogorov−Chaitin complexity. The exact definition (Ay et al. 2010) is that, given small constants $\delta \geq 0$ and $\Delta \geq 0$, the effective complexity EC[x] of any string $x \epsilon \mathcal{X}^*$ is defined as:

$$\text{EC}[x] := \text{Inf}\{\text{K}[\mathbb{E}] : x \text{ is typical for } \mathbb{E} \text{ and } \Sigma[\mathbb{E}] \leq K(x) + \Delta\}, \tag{209}$$

or as ∞ if this set is the empty set. The right-hand side of the above definition defines the minimization domain of the string x for effective complexity. Ensembles $\mathbb{E}$ of the minimization domain of $x \epsilon \mathcal{X}^*$ satisfy

$$\frac{K_U(x)}{1+\delta} <^+ \Sigma[\mathbb{E}] \leq K(x) + \Delta.$$

As Gell-Mann and Lloyd (2004) point out, it is often necessary to extend this definition of effective complexity by imposing additional constraints on the ensembles allowed in the minimization domain. These additional constraints can refer to certain properties of the string x that are judged important from the standpoint of a general scientific theory, or they can involve additional information about the processes that generated x (Ay et al. 2010). Ay et al. (2010) prove several properties of EC[x], such as its finiteness, and they show that incompressible strings are effectively simple, which is desirable given the criticism of the algorithmic complexity concept from Section 3.2.1. They also show that strings exist that have effective complexity close to their length $|x|$. Finally, one can show that EC[x] is related to Bennett's logical depth (1988, see Section 3.2.1). If the effective complexity of a string x exceeds a certain threshold, then the string must have an extremely large depth (Ay et al. 2010).

Moreover, Duncan Foley recently presented an interesting re-phrased formalism based on Bayesian inference. The Bayesian formulation allows us to interpret effective complexity in terms of the minimum description length principle of Wallace and Boulton (1968) and Rissanen (1989, 2007) as a two-part code (see notes on facticity and effective complexity by Foley and Oliver, 2011). To apply the method of Bayesian inference, Foley regards the problem of assigning probabilities to joint ensembles $\mathbb{E}$ as hypotheses, and the target string x as data. In this case, Bayes' theorem can be written as

$$P(\mathbb{E}|x) = P(\mathbb{E})\frac{P(x|\mathbb{E})}{P(x)},$$

where $P(\mathbb{E})$ is the prior probability assigned to the joint ensemble $\mathbb{E}$, $P(x|\mathbb{E})$ is the probability of the data given the ensemble ("likelihood"), and $P(x)$ is a normalizing constant. $P(\mathbb{E}|x)$ is the posterior probability of the joint ensemble given the data string x. Given the prior probability distribution $P(\mathbb{E}) = 2^{-K_U[\mathbb{E}]}$, the posterior distribution will be

$$\begin{aligned} P(\mathbb{E}|x) &\propto 2^{-K_U[\mathbb{E}]}P(x|\mathbb{E}) \\ &\propto 2^{-K_U[\mathbb{E}]}E[x]. \end{aligned}$$

The term $E[x]$ denotes the expected value of the corresponding discrete sequence. When we take the logarithm to base 2 to express information content in bits, we have

$$\log_2 P(\mathbb{E}|x) \propto -K_U[\mathbb{E}] + \log_2 E[x].$$

From Shannon's source coding theorem (Eq. 207), we know that the quantity $-\log_2 E[x]$ is the prefix code $d(x)$ with expected length $L[d(x), X]$ assigned to the data string x as a message to minimize average code length when the probabilities of messages are given by the joint ensemble $\mathbb{E}$. The negative logarithm of the posterior probability of a joint ensemble can therefore be regarded as the sum of the number of bits required to encode the ensemble as a program on U and as the length of code required to identify the string x given the distribution corresponding to $\mathbb{E}$. The logarithm of the posterior probability can also be interpreted in terms of the minimum description length principle from Sect 3.2.2 as the negative of the length of the two-part code transmitting the string x given the joint ensemble $\mathbb{E}$ as a generative model. Hence, we have the intuitive definition (Foley and Oliver 2011):

$$\mathrm{EC}[x] := K_U\left[\widehat{\mathbb{E}}_x = \arg\min_{\mathbb{E}}\{K_U[\mathbb{E}] - \log_2 E[x]\}\right].$$

It is important to note that this direct definition is a limited concept of effective complexity, as the information entropy $H[\mathbb{E}]$ of the ensemble is not evaluated.

In spite of its convincing concept and its conformity with the aforementioned expectations for a consistent complexity measure, in the following we will not consider the effective complexity in evaluating the emergent complexity of PD projects, as it is not computable. As Gell-Mann says: "There can exist no procedure for finding the set of all regularities of an entity" (Gell-Mann 1995, p. 2). This severe practical limitation leaves us with information-theoretic quantities based on dynamic entropies of joint ensembles that possess many (though not all) of the theoretically desired properties and can be efficiently and robustly estimated from data in a product development environment. These quantities will be discussed in the next chapter.

3.2.4 *Effective Measure Complexity and Forecasting Complexity*

Motivated in part by the theoretical weaknesses of the concept of stochastic complexity that were cited in Section 3.2.2 and by the uncomputability of algorithmic measures, the German physicist Peter Grassberger (1986) developed a simple but highly satisfactory complexity theory. He posits that complexity is the amount of information required for optimal prediction. We will begin by analyzing why this concept is plausible, and then go on to look at how to develop measuring concepts and make them fully operational. In general, there is a limit to the accuracy of any prediction of a given sociotechnical system set by the characteristics of the system itself, e.g. the free will of the decision makers, spontaneous human error, limited precision of measurement, sensitive dependence on initial conditions, etc. Suppose we have a model that is maximally predictive, i.e. its predictions are at the theoretical limit of accuracy. Prediction is always a matter of mapping inputs to outputs. In our application context, the inputs are the encoded observations of single instances of task processing (encoding, for instance, the labor units required to finalize a specific component, open design issues that need to be addressed before design release, etc.) and the outputs are the expectations about the work remaining, as well as macroscopic key performance indicators such as the finishing time of the project phase. However, usually not all aspects of the entire past are relevant for making good predictions. In fact, if the task processing is strictly periodic with a predefined cycle time, one only needs to know which of the φ phases the work process is in. For a completely randomized work process with independent and identically distributed (iid) state variables, the past is completely irrelevant for predicting the future. Because of this "memorylessness," the clever, evidence-based estimates of an experienced project manager on average do not outperform naïve guesses of the outcome based on means. If we ask how much information about the past is relevant in these two extreme cases, the correct answers are $\log_2(\varphi)$ and 0, respectively. It is intuitive that these cases are of low complexity, and more informative dynamics "somewhere in between" must be assigned high complexity

values. In terms of Shannon's famous information entropy $H[.]$ the "randomness" of the output either is simply a constant (low-period deterministic process with small algorithmic complexity) or grows precisely linearly with the length (completely randomized process with large algorithmic complexity). Hence, it can be concluded that both cases share the feature that corrections to the asymptotic behavior do not grow with the size of the dataset (Prokopenko et al. 2009). Grassberger considered the slow approach of the entropy to its extensive limit as an indicator of complexity. In other words, the subextensive components growing less rapidly with time than a linear function are of special interest for complexity evaluation.

When dealing with a Markovian model, such as the VAR model of cooperative task processing formulated in Section 2.2, only the present state of work remaining is relevant for predicting the future (see Eq. 8), so the amount of information needed for optimal prediction is simply equal to the amount of information needed to specify the current state. More formally, any predictor g will translate the one-dimensional infinite past $X_{-\infty}^{-1} = (X_{-\infty}, X_{-\infty+1}, \ldots, X_{-1})$ into an effective state $S = g\left[X_{-\infty}^{-1}\right]$ and then make its prediction on the basis of S. This is true whether or not $g[\cdot]$ is formally a state-space model as we have formulated. The amount of information required to specify the effective state in the case of discrete-type random variables (or discretized continuous-type random variables) can be expressed by Shannon's information entropy $H[S]$ (Cover and Thomas 1991). We will return to this point later in the chapter and take $H[S]$ to be the statistical complexity C_{GCY} of $g[\cdot]$ under the assumption of a minimal maximally predictive model of the stationary stochastic process $\{X_t\}$ ($t \in \mathbb{Z}$, see Eq. 228).

Shannon's information entropy represents the average information content of an outcome. Formally, it is defined for a discrete-type random variable X with values in the alphabet $\mathcal{X}$ and probability distribution $P(.)$ as

$$H[X] := -\sum_{x\epsilon\mathcal{X}} P(X = x)\log_2 P(X = x). \tag{210}$$

The information entropy $H[.]$ is non-negative and measures in [bits] the amount of freedom of choice in the associated decision process or, in other words, the degree of randomness. If we focus on the set $\mathcal{M}$ of maximally predictive models, we can define what Grassberger called "the true measure complexity C_μ of the process" as the minimal amount of information needed for optimal prediction:

$$C_\mu := \min_{g\in\mathcal{M}} H\left[g\left[X_{-\infty}^{-1}\right]\right]. \tag{211}$$

The true measure complexity is also termed "forecasting complexity" (Zambella and Grassberger 1988), because it is defined on the basis of maximally predictive models requiring the least average information content of the memory variable. We will use the term "forecasting complexity" in the following, as it is well-established and more intuitive. Unfortunately, Grassberger provided no procedure for finding the maximally predictive models or for minimizing the information content. However, he did draw the following conclusion. A basic result of information theory,

called "the data-processing inequality" (Cover and Thomas 1991), states that for any pair of random variables X and Y (or pair of sequences of random variables) the mutual information $I[.;.]$ follows the rule

$$I[X;Y] \geq I[g[X];Y].$$

It is therefore impossible to extract more information from observations by processing than was in the sample to begin with. Since the state S of the predictor is a function of the past, it follows that

$$I\left[X_{-\infty}^{-1};X_0^{\infty}\right] \geq I\left[g\left[X_{-\infty}^{-1}\right];X_0^{\infty}\right],$$

where $X_0^{\infty} = (X_0, X_1, \ldots, X_{\infty})$ represents the infinite future of the stochastic process including the "present" that is encoded in the observation X_0.

The mutual information $I[.;.]$ is another key quantity of information theory (Cover and Thomas 1991). It can be equivalently expressed on the basis of the joint $P(.,.)$ and marginal probability mass functions $P(.)$ as

$$I[X;Y] := \sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(X=x, Y=y) \log_2 \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)} \tag{212}$$

or in terms of the information entropy $H[.]$ as

$$\begin{aligned} I[X;Y] &= H[X] - H\left[X\middle|Y\right] \\ &= H[Y] - H\left[Y\middle|X\right] \\ &= H[X] + H[Y] - H[X,Y] \\ &= H[X,Y] - H\left[X\middle|Y\right] - H\left[Y\middle|X\right]. \end{aligned}$$

In the above equations, with the conditional entropy (also called equivocation, Cover and Thomas 1991) we have used another important information-theoretic quantity which measures the amount of information for the random variable X given the value of another random variable Y. It can be explicitly written as

$$H\left[X\middle|Y\right] = H[X,Y] - H[Y]. \tag{213}$$

The mutual information $I[.;.]$ is non-negative and measures the amount of information that can be obtained about one random variable by observing another. It is symmetric in terms of these variables. System designers often maximize the amount of information $I[A;B]$ shared by transmitted and received signals by choosing the best transmission technique. Channel coding guarantees that reliable communication is possible over noisy communication channels, if the rate of information transmission is below a certain threshold that is termed "the channel capacity," defined as the maximum mutual information for the channel over all possible

probability distributions of the signal (see Cover and Thomas 1991). According to Polani et al. (2006) mutual information should not be regarded as something that is transported from a transmitter to a receiver as a "bulk" quantity. Instead, the mutual information makes it possible to evaluate the intrinsic dynamics that can provide deeper insights into the inner structure of information; maximization of information transfer through selected channels appears to be one of the main evolutionary processes (Bialek et al. 2001; Polani et al. 2006).

In a similar manner, the conditional mutual information $I[X;Y|Z]$ (Cover and Thomas 1991) can be defined on the basis of the joint $P(.,.)$, marginal $P(.)$ and conditional $P(.|.)$ probability mass functions as

$$I[X;Y|Z]:=\sum_{z\epsilon\mathcal{Z}}P(Z=z)\sum_{x\epsilon\mathcal{X}}\sum_{y\epsilon\mathcal{Y}}P(X=x,Y=y|Z=z)\log_2\frac{P(X=x,Y=y|Z=z)}{P(X=x|Z=z)P(Y=y|Z=z)}. \tag{214}$$

The conditional mutual information can be interpreted in its most basic form as the expected value of the mutual information of two random variables given the value of a third one. Alternatively, we can write

$$I[X;Y|Z]=H[X|Z]+H[Y|Z]-H[X,Y|Z]. \tag{215}$$

Presumably, for optimal predictors, the amounts of information $I\left[X_{-\infty}^{-1};X_0^{\infty}\right]$ and $I\left[g\left[X_{-\infty}^{-1}\right];X_0^{\infty}\right]$ are equal and the predictor's state is just as informative as the original data. This is the case for so-called "ε-machines," which are analyzed below. Otherwise, the model would be missing potential predictive power. Another basic inequality is that $H[X]\geq I[X;Y]$, i.e. no variable contains more information about another than it does about itself (Cover and Thomas 1991). Even for the maximally predictive models it therefore holds that $H\left[X_{-\infty}^{-1}\right]\geq I\left[X_{-\infty}^{-1};X_0^{\infty}\right]$. Grassberger called the latter quantity $I\left[X_{-\infty}^{-1},X_0^{\infty}\right]$— the mutual information between the infinite past and future histories of a stochastic process—the effective measure complexity (EMC):

$$\text{EMC}:=I\left[X_{-\infty}^{-1};X_0^{\infty}\right]. \tag{216}$$

Recall that EMC is defined with reference to infinite sequences of random variables and is therefore only valid for stationary stochastic processes. The same is true for the forecasting complexity. For the sequence $(\ldots,X_{-1},X_0,X_1,\ldots)$ stationarity implies that the joint probability distribution $P(.,\ldots,.)$ associated with any finite block of n variables $X^n:=X_{t+1}^{t+n}=(X_{t+1},\ldots,X_{t+n})$ is independent of t and only depends on the block length n. The independency of the joint probability distribution of t can limit the evaluation of PD projects in industry, as the dynamical dependencies between process and product can significantly change over time. In this case an alternative complexity measure—known as the "binding

information"—developed by Abdallah and Plumbley (2012) should be taken into consideration, as it can be used to evaluate non-stationary processes of different kinds.

If optimal predictions of the stationary stochastic process are influenced by events in the arbitrarily distant past, the mutual information diverges and the measure EMC tends to infinity (see discussion of predictive information I_{pred} below).

Shalizi and Crutchfield (2001) proved that the forecasting complexity gives an upper bound of the EMC:

$$\text{EMC} \leq C_{\mu}. \tag{217}$$

In terms of a communication channel, EMC is the effective information transmission rate of the process. The units are bits. C_{μ} is the memory stored in that channel. Hence, the inequality above means that the memory needed to carry out an optimal prediction of the future cannot be less than the information that is transmitted from the past $X_{-\infty}^{-1}$ to the future X_0^{∞} (by storing it in the present). However, the specification of how the memory has to be designed and managed cannot be derived on the basis of information-theory considerations. Instead, a constructive and more structural approach based on a theory of computation must be developed. A highly satisfactory theory based on "causal states" was developed by Crutchfield and Feldman (2003). These causal states lead to the cited ε-machines, as well as the Grassberger–Crutchfield–Young statistical complexity C_{GCY}, which will be presented later in this chapter.

EMC can be estimated purely from historical data, without use of a generative stochastic model of cooperative work. If the data is generated by a model in a specific class but with unknown parameter values, we can derive closed-form solutions for EMC, as will be shown in Sections 4.1.1, 4.1.2 and 4.1.3 for a VAR (1) model (cf. Eq. 262). The mutual information between the infinite past and future histories of a stochastic process has been considered in many contexts. It is termed, for example, excess entropy **E** (Crutchfield and Feldman 2003; Ellison et al. 2009; Crutchfield et al. 2010), predictive information $I_{pred}(n \to \infty)$ (Bialek et al. 2001, see below), stored information (Shaw 1984), past-future information I_{p-f} (Li and Xie 1996, see Section 5.1) or simply complexity (Arnold 1996; Li 1991). Rissanen (1996, 2007) also refers to the part of stochastic complexity required for coding model parameters as model complexity. Hence, there should be a close connection between Rissanen's ideas of encoding a data stream based on generative models and Grassberger's ideas of extracting the amount of information required for optimal prediction. In fact, if the data allows a description by a model with a finite number of independent parameters, then mutual information between the data and the parameters is of interest, and this is also the predictive information about all of the future (Bialek et al. 2001). Rissanen's approach was further strengthened by a result put forward by Vitányi and Li (2000) showing that an estimation of parameters using the MDL principle is equivalent to Bayesian parameter estimations with

a "universal" prior (Li and Vitányi 1997). Since the mutual information between the infinite past and future histories can quantify the statistical dependency structures of cooperative work processes, it will be used in the following to evaluate the emergent complexity in PD projects.

In addition to C_μ and EMC, another key invariant of stochastic processes that was discovered much earlier is Shannon's source entropy rate (Cover and Thomas 1991):

$$h_\mu := \lim_{\eta \to \infty} \frac{H[X^{n=\eta}]}{\eta}. \tag{218}$$

This limit exists for all stationary processes. The source entropy rate is the intrinsic randomness that cannot be reduced, even after considering statistics over longer and longer blocks of generating variables. The unit of h_μ is bits/symbol. It is also known as per-symbol entropy, thermodynamic entropy density, Kolmogorov–Sinai entropy or metric entropy. The source entropy rate is zero for periodic processes. Surprisingly, it is also zero for deterministic processes with infinite memory. The source entropy rate is larger than zero for irreducibly unpredictable processes like the cited iid process or Markov processes. The capacity of a communication channel must be larger than h_μ for error-free data transmission (Cover and Thomas 1991). Interestingly, the source entropy rate is related to the algorithmic complexity (Section 3.1): h_μ is equal to the average length (per variable) of the minimal program with respect to U that, when run, will cause the Universal Turing Machine to produce a typical configuration and then halt (Cover and Thomas 1991). In the above definition the variable $H[X^n]$ is the joint information entropy of length-n blocks $(X_{t+1}, \ldots, X_{t+n})$. This entropy is not the entropy of a finite string x^n with length n; rather, it is the entropy of sequences with length n drawn from mainly much longer or infinite output generated by the process in the steady state. The variable n is the nonnegative order parameter and can be interpreted as an expanding observation window of length n over the output. In the following, we will use the shorthand notation $H(n)$ to represent this kind of entropy, which is also termed Shannon block entropy (Grassberger 1986; Bialek et al. 2001). For discrete-type random variables the block entropy is defined as

$$\begin{aligned} H(n) &:= H[X^n] \\ &= H[X_{t+1}, \ldots, X_{t+n}] \\ &= -\sum_{\mathcal{X}} \cdots \sum_{\mathcal{X}} P\big(X_{t+1} = x_{j(t+1)}, \ldots, X_{t+n} = x_{j(t+n)}\big) \\ &\quad \cdot \log_2 P\big(X_{t+1} = x_{j(t+1)}, \ldots, X_{t+n} = x_{j(t+n)}\big) \end{aligned} \tag{219}$$

with

$$H(0) := 0. \tag{220}$$

The sums in Eq. 219 run over all possible blocks of length n. The corresponding definition for continuous-type variables will be given in Eq. 233. Interestingly, the

length-n approximation $h_\mu(n)$ of the entropy rate h_μ can be defined as the two-point slope of the block entropy $H(n)$:

$$h_\mu(n) := H(n) - H(n-1), \tag{221}$$

with

$$h_\mu(0) := \log_2|\mathcal{X}|. \tag{222}$$

Vice versa, $h_\mu(n)$ is the discrete derivative of the block entropy with respect to the block length n. In this sense, the length-n approximation is a dynamic entropy representing the entropy gain (Crutchfield and Feldman 2003). It can be seen that the entropy gain can also be expressed as conditional entropy

$$h_\mu(n) := H\left[X_n \middle| X^{n-1}\right].$$

In the limit of infinitely long blocks, it is equal to the source entropy rate

$$h_\mu = \lim_{\eta\to\infty} h_\mu(n = \eta). \tag{223}$$

In general $h_\mu(n)$ differs from the estimate $H(n)/n$ for any given n but converges to the same limit, namely the source entropy rate h_μ. According to Crutchfield and Feldman (2003), $h_\mu(n)$ typically overestimates h_μ at finite n, and each difference $h_n - h_\mu$ represents the difference between the entropy rate conditioned on n measurements and the entropy rate conditioned on an infinite number of measurements. As such, it estimates the information-carrying capacity in blocks in which the difference is not actually random but arises from correlations. The difference $h_n - h_\mu$ can therefore be interpreted as the local predictability. These local "overestimates" can be used to define a universal learning curve $\Lambda(n)$ (Bialek et al. 2001) as

$$\Lambda(n) := h_\mu(n) - h_\mu, \qquad n \geq 1. \tag{224}$$

EMC is simply the discrete integral of $\Lambda(n)$ with respect to the block length n, which controls the speed of convergence of the dynamic entropy to its limit (Crutchfield et al. 2010):

$$\text{EMC} := \sum_{n=1}^{\infty} \Lambda(n). \tag{225}$$

In the sense of a learning curve, EMC measures the amount of apparent randomness at small block length n that can be "explained away" by considering correlations between blocks with increasing lengths $n+1, n+2, \ldots$. Grassberger (1986) analyzed the manner in which $h_\mu(n)$ approaches its limit h_μ, noting that for certain classes of stochastic processes with long-range correlations, the convergence can be very

slow and that this is an indicator of complexity. He also found that the approach of the limit can be so slow that $h_\mu(n)$ decays slower than $1/n$ and therefore EMC is infinite. These processes are termed infinitary processes (Travers and Crutchfield 2014). When EMC is infinite, then the manner of its divergence can provide additional information of how a system's internal state space is coarse grained (see e.g. Bialek et al. 2001 and Crutchfield and Feldman 2003). This phenomenon has been analyzed in greatest detail by Bialek et al. (2001). To carry out their analysis, they defined the predictive information $I_{pred}(n)\,(n \geq 1)$ as the mutual information between a block of length n and the infinite future following the block:

$$\begin{aligned} I_{pred}(n) &:= \lim_{\eta \to \infty} I\left[X_{-n}^{-1}; X_0^{\eta}\right] \\ &= \lim_{\eta \to \infty} H(n) + H(\eta) - H(n+\eta). \end{aligned} \tag{226}$$

Bialek et al. (2001) showed that even if $I_{pred}(n)$ diverges as n tends to infinity, the way in which it grows is an indicator of a process's complexity in its own right. They also emphasized that the predictive information is the subextensive component of the entropy:

$$H(n) = nh_\mu + I_{pred}(n). \tag{227}$$

From the above equation, it can be seen that the sum of the first n terms of the discrete integral of the universal learning curve $\Lambda(n)$, that is, $H(n) - nh_\mu$, is equal to $I_{pred}(n)$ (Abdallah and Plumbley 2012):

$$I_{pred}(n) = \sum_{i=1}^{n} \Lambda(i).$$

As expected, $I_{pred}(n)$ (as well as EMC) is zero for an iid process. According to Bialek et al. (2001), it is positive in all other cases and grows less rapidly than a linear function (subextensive). $I_{pred}(n)$ may either stay finite or grow infinitely. If it stays finite, no matter how long we observe the past of a process, we gain only a finite amount of information about the future. This holds true, for instance, for the cited periodic processes after the period φ has been identified. A longer period results in larger complexity values and $I_{pred}(n \to \infty) = \text{EMC} = \log_2(\varphi)$. For some irregular processes, the best predictions may depend only on the immediate past, e.g. in our Markovian model of task processing or generally when evaluating a system far away from phase transitions or symmetry breaking. In these cases, $I_{pred}(n \to \infty) = \text{EMC}$ is also small and is bound by the logarithm of the number of accessible states. Systems with more accessible states and larger memories are assigned larger complexity values. On the other hand, if $I_{pred}(n)$ diverges and optimal predictions are influenced by events in the arbitrarily distant past, then the rate of growth may be slow (logarithmic) or fast (sublinear power). If the acquired data allows us to infer a model with a finite number of independent parameters, or to identify a set of generative rules that can be described by a finite

number of parameters, then $I_{pred}(n)$ grows logarithmically with the size of the sample. The coefficient of this divergence counts the dimensionality of the model space (i.e. the effective number of independent parameters). Sublinear power-law growth can be associated with infinite parameter models or with nonparametric models, such as continuous functions with smoothness constraints. Typically these cases occur where predictability over long time scales is governed by a progressively more detailed description as more data points are observed.

To make the previously introduced key invariant C_μ (forecasting complexity, Eq. 211) of a stochastic process operational in terms of a theory of computation and to clarify its relationship to the other key invariant EMC (effective measure complexity, Eq. 225) by using a structurally rich model and not simply a purely mathematical representation of a communication channel, in the following we refer to the seminal work of Crutchfield and Young (1989, 1990) on computational mechanics. They provided a procedure for finding the minimal maximally predictive model and its causal states by means of an ε-machine (Ellison et al. 2009; Crutchfield et al. 2010). The general goal of building an ε-machine is to find a constructive representation of a nontrivial process that not only allows good predictions on the basis of the stored predictive information, but also reveals the essential mechanisms that produce a system's behavior. To build a minimal maximally predictive model of a stationary stochastic process, we can formally define an equivalence relation $x_{-\infty}^{-1} \sim \widehat{x}_{-\infty}^{-1}$ that groups all process histories that give rise to the same prediction:

$$x_{-\infty}^{-1} \sim \widehat{x}_{-\infty}^{-1} :\Leftrightarrow \left\{P\left(X_0^{\infty} \middle| X_{-\infty}^{-1} = x_{-\infty}^{-1}\right) = P\left(X_0^{\infty} \middle| X_{-\infty}^{-1} = \widehat{x}_{-\infty}^{-1}\right)\right\}.$$

Hence, for the purpose of forecasting, two different sequences of past observations are considered equivalent if they result in the same predictive distribution. The above equivalence relation determines the process's causal state, which partitions the space $X_{-\infty}^{-1}$ of pasts into sets that are predictively equivalent. The causal state $\varepsilon\left(x_{-\infty}^{-1}\right)$ of $x_{-\infty}^{-1}$ is its equivalence class

$$\varepsilon\left(x_{-\infty}^{-1}\right) := \left\{\widehat{x}_{-\infty}^{-1} : x_{-\infty}^{-1} \sim \widehat{x}_{-\infty}^{-1}\right\},$$

and the causal state function $\varepsilon(.)$ defines a deterministic sufficient memory $\mathcal{M}_\varepsilon$ (see Shalizi and Crutchfield 2001; Löhr 2012). The set of memory states of the ε-machine is simply the set of causal states

$$\mathcal{M}_\varepsilon := \left\{\varepsilon\left(x_{-\infty}^{-1}\right) : x_{-\infty}^{-1} \in \mathcal{X}^{\mathbb{N}}\right\}.$$

The set $\mathcal{X}$ represents the finite alphabet on which the stationary stochastic process is defined. The set of causal states $\mathcal{M}_\varepsilon$ does not need to be countable and can therefore represent either discrete or continuous state spaces. Shalizi and Crutchfield (2001) showed that the equivalence relation $x_{-\infty}^{-1} \sim \widehat{x}_{-\infty}^{-1}$ is minimally sufficient and unique. Hence, it allows the highest compression of the data, while containing all the relevant information on local dynamics. For practical purposes, longer and

longer histories are analyzed, from x_{-L}^{-1} up to a predefined maximum length $L = L_{\max}$, and the partition into classes for a fixed future horizon X_0^t is obtained. In principle, we start at the most coarse-grained level, grouping together those histories that have the same predictive distribution for the next observable X_0, and then refine the partition. The refinement is recursively carried out by further subdividing the classes using the predictive distributions of the next two observables X_0^1, the next three observables X_0^2, etc.

After all causal states have been identified, an ε-machine can be constructed. To simplify the definition of the forecasting complexity C_μ, we start by using an informal representation in the form of a stochastic output automaton that is expressed by the causal state function ε, a set of transition matrices $\mathcal{T}$ for the states defined by ε, and the start state s_0. The start state is unique. Given the current state $s \in \mathcal{M}_\varepsilon$ of the automaton, a transition to the next state $s' \in \mathcal{M}_\varepsilon$ is determined by the output symbol (or measurement) $x \in \mathcal{X}$. State-to-state transitions are probabilistic and must therefore be represented for each output symbol x by a separate transition matrix $T^{(x)} \in \mathcal{T}$. Each row and column of the transition matrices in the set $\mathcal{T}$ stands for an individual causal state. A stochastic output automaton can also be transformed into an equivalent edge-emitting hidden Markov model (Löhr 2012). A hidden Markov model is a universal machine that is defined over a set of non-observable internal states $\mathcal{M}_\varepsilon$. It therefore does not directly reveal its internal mechanisms to external observers; it only expresses them indirectly through emitted symbols. The emitted symbols are edge-labels of the hidden states. The model can be formally represented by the tuple $\left(\mathcal{M}_\varepsilon, \mathcal{X}, \pi, \left\{T^{(x)}\right\}\right)$. The start state of the hidden Markov model is not unique but determined by an initial probability distribution π. Depending on the current internal state s_t, at each time step t a transition to the new internal state s_{t+1} is made and an output symbol x_{t+1} from the alphabet $\mathcal{X}$ is emitted. The corresponding entry $T_{ij}^{(x)}$ of the transition matrix $T^{(x)}$ gives the probability $P\left(S_{t+1} = s_{t+1}, X_{t+1} = x_{t+1} \middle| S_t = s_t\right)$ of transitioning from current state s_t indexed by i to the next s_{t+1} indexed by j on "seeing" measurement x. This operation may also be thought of as a weighted random walk on the associated graphical model (Travers and Crutchfield 2011): from the current state s_t, the next state s_{t+1} is determined by selecting an outgoing edge from current state s_t according to their probabilities. After a transition has been selected, the model moves to the new state and outputs the symbol of the current state x labeling the edge. The transition matrices are usually non-symmetric. From the theory of Markov processes (see e.g. Puri 2010) it is well known that in a steady state the probability distribution over the hidden states is independent of the initial-state distribution. Edge-emitting hidden Markov models can also be expressed by an initial probability distribution π, by a state process $\{S_t\}$ and by an output process $\{X_t\}$, which means that they are theoretically similar to the continuous-type linear dynamical systems that were analyzed in Section 2.9. However, continuous-type linear dynamical systems usually do not possess the property of "unifilarity" (see below) and therefore cannot be used to directly calculate the entropy rate of the process.

To obtain the transition matrices $T^{(x)}$, one can parse the data sequence of interest in a sequential manner, identify all causal state transitions defined by ε over histories x_0^t and x_0^{t+1}, and estimate the transition probabilities $P\left(S', X = x_{t+1} \middle| S\right)$ using frequency counting (MLE, see Section 2.4) or Bayesian methods. The transition probabilities allow calculation of an invariant probability distribution $P(S)$ over the causal states. This probability is obtained as the normalized principal eigenvector of the transition matrix $T = \Sigma_{x \in \mathcal{X}} T^{(x)}$ (Ellison et al. 2009). The matrix T is stochastic and $\Sigma_{j=1}^{|\mathcal{M}_\varepsilon|} T_{ij} = 1$ holds for each i.

Interestingly, causal states have a Markovian property in that they render the past and future statistically independent. In other words, they shield the future from the past:

$$P\left(X_{-\infty}^{-1}, X_0^{\infty} \middle| S\right) = P\left(X_{-\infty}^{-1} \middle| S\right) P\left(X_0^{\infty} \middle| S\right).$$

Moreover, they are optimally predictive in the sense that knowing what causal state a process is in is as good as having the entire past: $P\left(X_0^{\infty} \middle| S\right) = P\left(X_0^{\infty} \middle| X_{-\infty}^{-1}\right)$. Causal shielding is therefore equivalent to the fact that the causal states capture all of the information shared between past and future. Hence, $I\left[S, X_0^{\infty}\right] = \text{EMC}$. Out of all maximally predictive models $\mathcal{M}$ for which $I\left[\mathcal{M}, X_0^{\infty}\right] = \text{EMC}$, the ε-machine captures the minimal amount of information that a stationary stochastic process must store in order to communicate all excess entropy from the past to the future. Accordingly, the ε-machine is as close to perfect determinism as any rival that has the same predictive power (Jänicke and Scheuermann 2009). The minimal amount of information that must be stored on a stationary stochastic process $X_{-\infty}^{\infty} = (\ldots, X_{-1}, X_0, X_1, \ldots)$ for optimal prediction is the Shannon information entropy over the stationary distribution of its ε-machine's causal states—the forecasting complexity—and it holds that

$$C_\mu\left(X_{-\infty}^{\infty}\right) = H[S].$$

Because of its significance in complex systems science, the forecasting complexity is also termed Grassberger–Crutchfield–Young statistical complexity C_{GCY} (Shalizi 2006). It should not be confused with Rissanen's stochastic complexity C_{SC} from Eq. 208, because the underlying concepts are based on a theory of computation. We have (Ellison et al. 2009)

$$\begin{aligned} C_{GCY} &= H[S] \leq H[\mathcal{M}] \\ C_{GCY} &= -\sum_{s \epsilon \mathcal{M}_s} P(S) \log_2 P(S) \leq H[\mathcal{M}]. \end{aligned} \tag{228}$$

As we have argued, the causal states are an objective property of the stochastic process under consideration and therefore the associated statistical complexity C_{GCY} cannot be influenced by our ineptness as modelers or our (possibly poor)

means of description. It is equal to the length of the shortest description of the past that is relevant to the actual dynamics of the system. As was shown above, for iid sequences it is exactly 0, and for periodic sequences it is $\log_2(\varphi)$. A detailed description of an algorithm providing an ε-machine reconstruction and calculation of C_{GCY} for one-dimensional and two-dimensional time series can be found in Shalizi and Shalizi (2004, 2003).

Moreover, the entropy rate h_μ can be directly calculated on the basis of a process's ε-machine (Ellison et al. 2009) because of unifilarity:

$$\begin{aligned} h_\mu &= H\left[X\middle|S\right] \\ &= -\sum_{s\epsilon\mathcal{M}_s} P(S) \sum_{xs'\epsilon\mathcal{X}\mathcal{M}_s} T_{ss'}^{(x)} \log_2 \sum_{s'\epsilon\mathcal{M}_s} T_{ss'}^{(x)} . \end{aligned}$$

$\mathcal{X}\mathcal{M}_s$ denotes the set whose elements are generated by concatenating all elements of the sets $\mathcal{X}$ and $\mathcal{M}_s$. Unifilarity means that from the start state s_0 of the process, each generated sequence of observations corresponds to exactly one sequence of causal states. In a hidden Markov model representation of an ε-machine this property can easily be verified. For each hidden state, each emitted symbol appears on at most one edge. In the above equation, we used the shorthand notation $T_{ss'}^{(x)}$ to denote the matrix entry $T_{ij}^{(x)}$ corresponding to causal state s in row i and causal state s' in column j of the transition matrix associated with output symbol x. The probability $P(S)$ denotes the asymptotic probability of the causal states.

In a recent paper, Gu et al. (2012) extended the framework of ε-machines by allowing the casual states to have quantum mechanical properties. This extension also makes it possible to define the quantum complexity of a stochastic process. Interestingly, the quantum complexity of a process is bounded below by EMC and above by C_{GCY} (Wiesner 2015).

An especially interesting variant of Grassberger's classic definition of the effective measure complexity has recently been developed by Ball et al. (2010). These authors also quantify strong emergence within an ensemble of histories of a complex system in terms of mutual information between past and future history, but focus on the part of the information that persists across an interval of time $\tau > 0$. As such, we can specify the "persistent mutual information" as a complexity measure in its own right that evaluates the deficit in the information entropy in the joint history compared with that of past and future taken independently. Formally, the persistent mutual information can be defined on the basis of the EMC (Eq. 216) extended by the lead time τ to evaluate the persistent part as

$$\mathrm{EMC}(\tau) := I\left[X_{-\infty}^{-1}; X_{\tau}^{\infty}\right], \tag{229}$$

where $X_{-\infty}^{-1}$ designates the history of the stochastic process from an infinite past to the present, and X_{τ}^{∞} is the corresponding future of the system from the later time

τ onwards. The key distinguishing feature of the definition above is that it ignores the information captured in block $X_0^{\tau-1}$, that is, the intervening interval of observations of length τ. For continuous state variables, EMC(τ) has the merit of being independent of continuous changes of the variable, as long as they preserve time labeling (Ball et al. 2010). EMC(τ) is known to be a Lyapunov function for the process, so that it decays with increasing lead time (Ay et al. 2012). For positive lead times the persistent mutual information is nonzero if a process has a memory mechanism to store the predictive information persistently and is therefore sensitive to how a system's state space is observed (Marzen and Crutchfield 2014). Li (2006) defines an information-regular process as a process whose persistent mutual information converges to zero as the lead time grows over all given limits and it holds that $\mathrm{EMC}(\tau) \to 0$ as $\tau \to \infty$. Otherwise, the process is information-irregular. The differences between the effective measure complexity and the persistent mutual information for continuous-state processes are presented in more detail in Section 4.1.6.

It is evident that the persistent mutual information enables the specification of an intuitive lower bound on EMC:

$$\mathrm{EMC}(\tau) \leq \mathrm{EMC}. \tag{230}$$

In fact, for zero lead time we have

$$\mathrm{EMC}(0) = \mathrm{EMC}.$$

The recent work of James et al. (2011), Marzen and Crutchfield (2014) and others has shown that a fine decomposition of the persistent mutual information can be carried out, essentially breaking it down into two pieces. With respect to emergent complexity, the most interesting piece is the so-called "elusive information" $\sigma_\mu(\tau)$, which is the mutual information between the past $X_{-\infty}^{-1}$ and the future X_τ^∞ conditioned on the length-τ present $X_0^{\tau-1}$ (cf. Eq. 214):

$$\sigma_\mu(\tau) := I\left[X_{-\infty}^{-1}; X_\tau^\infty \middle| X_0^{\tau-1}\right]. \tag{231}$$

According to the analysis by James et al. (2011) the elusive information has an especially interesting interpretation: it represents the Shannon information that is communicated from the past to the future, but does not flow through the currently observed length-τ sequence $X_0^{\tau-1}$. The key distinguishing feature of the persistent mutual information is that it is nonzero for positive length τ if a process necessarily has hidden states. In this case, all the information from the past that is relevant for generating future behavior has to be stored by an internal configuration to arrive at a complete description of the process. The internal configuration is necessary to keep track of the state information, because the present sequence of observations $X_0^{\tau-1}$ can only capture features of shorter term correlation and therefore does not have enough capacity to capture all the features that are relevant for forecasting. In the

words of James et al. (2011): "This is why we build models and cannot rely on only collecting observation sequences." For instance, for the n-th order Markov chains that were introduced in Section 3.2.2, we have $\sigma_\mu(\tau) = 0$ for lead times τ that are larger than or equal to the model order n. In this case with only fully observable state variables, the length-n memory of the chain model serves as the effective state, rendering the process's past and future independent (Marzen and Crutchfield 2014). For infinite-order Markov chains EMC(τ) only vanishes asymptotically. Therefore, the elusive information is sensitive to which extent a system's internal state space is coarse grained (Marzen and Crutchfield 2014).

3.3 Complexity Measures from Theories of Systematic Engineering Design

The most prominent complexity theory in the field of systematic engineering design has been developed by Suh (2005). His theory aims at providing a systematic way of designing products and large-scale systems, as well as of determining the best designs from those proposed. Suh's complexity theory is based on his famous axiomatic design theory (Suh 2001). He defines complexity in the functional domain rather than in the physical domain of the design world. In the functional domain, uncertainty is measured through information-theoretic quantities like the information content that was already introduced and defined in Section 3.2.2. Alternative approaches to characterizing complexity in engineering design that are not based on information-theory and statistical models (see e.g. Lindemann et al. 2009; Kreimeyer and Lindemann 2011) are only very briefly addressed in the following, as they tend to be valid only for evaluating structural and not time-dependent complexity.

In Suh's axiomatic design theory, the product to be developed and the problem of solving the design issues are coupled through functional requirements (FRs) and design parameters (DPs). He proposes two axioms for design: the independence axiom and the information axiom. The independence axiom states that the FRs should be maintained by the designer or design team independent of each other. When there are two or more FRs, the design solution must be such that each of the FRs can be satisfied without affecting any of the other FRs. This means that a correct set of DPs is to be chosen so as to satisfy the FRs and maintain their independence. If the independence can be maintained for all FRs, the design is said to be "uncoupled." An uncoupled design is an optimal solution in the sense of the theory. Once the FRs are established, the next step in the design process is the conceptualization process, which occurs during the mapping process from the functional to the physical domain.

The conceptualization process may produce several designs, all of which may be satisfactory in terms of the independence axiom. Even for the same task defined by a given set of FRs, it is likely that different engineers will come up with different

designs, because there are many solutions that satisfy a given set of m FRs $(FR_1, \ldots, FR_m)$. The information axiom provides a quantitative measure of the merits of a given design, and is thus useful in selecting the best design from among those that are acceptable. The information axiom is formulated within an information-theory framework and states that the best design is that with the highest probability of success. Following the definition of the Shannon information content in Eq. 205 the information content I_i for a given functional requirement $FR_i (1 \leq i \leq m)$ is expressed as the logarithmic probability p_i of satisfying this specific FR:

$$\begin{aligned} I_i &= \log_2 \frac{1}{p_i} \\ &= -\log_2 p_i. \end{aligned}$$

In the general case of m specified FRs, the information content I_{sys} for the entire system under study is

$$I_{sys} = -\log_2 P(X^m),$$

where $P(X^m)$ denotes the joint probability that all m FRs are satisfied. When all FRs are statistically independent, as in an uncoupled design, the information content I_{sys} can be decomposed into independent summands and expressed as

$$\begin{aligned} I_{sys} &= \sum_{i=1}^{m} I_i \\ &= -\sum_{i=1}^{m} \log_2 p_i. \end{aligned}$$

When not all FRs are statistically independent (in the so called "decoupled design"), there holds

$$I_{sys} = -\sum_{i=1}^{m} \log_2 p_{i|\{j\}} \quad \text{for} \{j\} = \{1, \ldots, i-1\}$$

In the above equation $p_{i|\{j\}}$ is the conditional probability of satisfying FR_i given that all other correlated $\left\{FR_j\right\}_{j=1,\ldots,i-1}$ are also satisfied. It is assumed that the FRs are ordered according to their number of correlations. The information axiom states that the best design is that with the smallest I_{sys}, because the least amount of information in the sense of Shannon's theory is required to achieve the design goals. When all probabilities are one, the information content is zero and the design is optimal in the sense of the axiom. Conversely, when one or more probabilities are zero, the information required is infinite and the system has to be redesigned to satisfy the information axiom.

The probability of success p_i can be determined by the intersection of the design range defined by the designers to satisfy the FRs and the ability of the system to produce the part within the specified range. This probability can be computed by specifying the design range (r) for the FR and by determining the system range (sr) that the proposed design can provide to satisfy the FR. The lower bound of the specified design range for functional requirement FR_i is denoted by $r^l[FR_i]$, and the upper bound by $r^u[FR_i]$. The system range can be modeled in statistical terms on the basis of a probability density function (*pdf*, see Section 2.1). The *pdf* is specified over the theoretically feasible state space. The system *pdf* is denoted by $f_{sys}[FR_i]$. The overlap between the design and system ranges is called "the common range" (cr), and this is the only range where the FR is satisfied. Consequently, the area A_{cr} under the system *pdf* within the common range is the design's probability of achieving the specified goal. Hence, the information content I_i can be expressed as

$$\begin{aligned} I_i &= -\log_2 A_{cr} \\ &= -\log_2 \int_{r^l[FR_i]}^{r^u[FR_i]} f_{sys}[FR_i] \; dFR_i. \end{aligned}$$

Suh (2005) considers a design to be complex when its probability of success is low and hence the information content I_{sys} required to satisfy the FRs is high. Complex designs often arise when there are many components, because as their number increases through functional decomposition, the probability that some of them do not meet the specified requirements also increases, such as when the interfaces between components introduce additional errors. In order to steer the design process toward more effective, efficient and robust large-scale systems, a dedicated complexity axiom is defined that simply states "reduce the complexity of a system" (Suh 2005). The quantitative measure for complexity in the sense of this axiom is the information content, which was defined in the above equations. The rationale behind the axiom is that complex systems may require more information to make the system function. Therefore, Suh (2005) ties the notion of complexity to the design range for the FRs—the tighter the design range, the more difficult it becomes to satisfy the FRs. An uncoupled design is likely to be least complex. However, the complexity of a decoupled design can be high because of so-called "imaginary complexity" if we do not understand the system. It is not truly complex, but it appears to be so because of our lack of understanding of generalized or physical functions.

According to Suh (2005) complexity can also be a function of time if the system range changes as a function of time. In this case, we must differentiate between two types of time-dependent complexity: time-dependent combinatorial complexity and time-dependent periodic complexity. Time-dependent combinatorial complexity is defined as the complexity that increases as a function of time because of a continued expansion in the number of possible combinations of FRs and DPs in time, which may lead to chaotic behavior or system failure. It occurs because future events

occur randomly in time and only have a limited predictability, even though they depend on the current state. Conversely, periodic complexity is defined as the complexity that only exists in a finite time period, resulting in a finite and limited number of probable configurations. Concerning a system subjected to combinatorial complexity, Suh (2005) concludes that the uncertainty of future outcomes continues to grow over time, and as a result, the system cannot have long-term stability and reliability. In the case of systems with periodic complexity, it is assumed that the system is deterministic and can renew itself over each period. Therefore, he concludes that a stable and reliable system must be periodic. It is readily apparent that a system with time-dependent combinatorial complexity can be changed to one with time-dependent periodic complexity by defining a set of functions that repeat periodically. This can be achieved temporally, geometrically, thermally, electrically and by other constructive means. In conclusion, engineered systems in PD should have small time-independent real and imaginary complexities and no time-dependent combinatorial complexity. If the system range must change as a function of time, the developer should be able to introduce time-dependent periodic complexity. These criteria need to be satisfied regardless of the size of the system or the number of FRs and DPs specified for the system.

Although Suh's complexity theory is grounded in axiomatic design theory and has been successfully applied in different domains, our criticism is that product and design problems are evaluated irrespective of the work processes, which are needed to decompose the FRs and DPs. The decomposition is a highly cooperative process that must be taken into account to satisfy all specified FRs on time and to avoid cycles of continual revision. Furthermore, the fact that Suh uses the information content I_{sys} directly as a complexity measure can be a point of criticism. I_{sys} is a simple additive measure that only represents the encoded length of the design in terms of binary design decisions; it does not take into account the encoding scheme. However, both parts of the description of a design are important because the description can always be simplified by formulating more complicated design rules, more complex standard components or interfaces (cf. Section 3.2.2). Lastly, Suh (2005) does not define specific measures for time-dependent complexity.

El-Haik and Yang (1999) have extended Suh's theory by representing the imaginary part of complexity through the differential entropy (Chapter 4) associated with the joint *pdf* of FRs with three components of variability, vulnerability and correlation. These components evaluate the product design according to the vector of DPs (see Summers and Shah 2010). Although this approach can be used to assess the mapping from the FRs to the DPs through an analysis of the topological structure of the design structure matrix (Browning 2001, see discussion below) and the variability of the design parameters (measured by the differential entropy of the joint pdf of DPs), the dynamics of the development processes in terms of a work transformation matrix (WTM, Section 2.2) are not taken into account. An alternative view introduced by Braha and Maimon (1998) suggests that complexity is a fundamental characteristic of the information content within either the product or the process. They introduce two measures that quantify either the structural representation of the information or the functional probability of achieving the specified

requirements. The measures can be applied to compare products and processes at different levels of abstraction. The process is nominally defined as mapping between the product and problem, where the coupling determines process complexity. The size of the process is defined as the summation over the number of instances of operators (relationships) and operands (entities). A process instance is a sequence of the instances of operands and operators. The average information content of sequences can be evaluated on the basis of the block entropy (Eq. 219). As the design takes on different types of representations through the development stages, the average information contained changes. Braha and Maimon (1998) suggest that the ratio of the amount of average information content between the initial and current states is a measure of the current abstraction level. The effort required to move between abstraction levels is inversely proportional to this ratio. The proportionality constant is the information content of the current state. Summers and Shah (2010) follow these lines of reasoning and propose a process size complexity measure that includes the vocabulary of the specific representation for the problem, the product, the development process and the four operators available for sequencing the states of the design evolution. The measure is defined as

$$Cx_{size\_process} := \left(M^o + C^o + P_{op}\right)\ln\left(idv + ddv + dr + mg + a_{op} + e_{op} + s_{op} + r_{op}\right).$$

In the above definition the size of the vocabulary is represented by the total number of possible primitive modules (M^o), possible relations between these modules (C^o) and possible operators and operands (P_{op}). The additional parameters denote the variables whose values are controlled by the designer (idv), are derived from the independent design parameters, other dependent variables and design relations (ddv), are constraints that dictate the association between the other design variables (dr), or are used to determine how well the current design configuration meets the goals (mg), plus the four operators available for sequencing the states. Although the extended concepts based on information content within either the product or the process are appealing, the fact that the development process is only analyzed on stage-dependent hierarchical description levels, not on the basis of an explicit state-space model of cooperative work, opens it to criticism. Moreover, dynamic entropies in the sense of Grassberger's theory are not taken into account to evaluate time-dependent combinatorial complexity in an open organizational system. Last but not least, in real design problems, it is difficult to identify all operators and operands in advance and to specify valid sequences leading from one level of abstraction to the next.

In addition to methods for measuring characteristics of the design based on information-theoretic quantities, a large body of literature has been published on the design structure matrix (Steward 1981) as a dependency modeling technique supporting complexity management by focusing attention on the elements of a system and the dependencies through which they are related. Recent surveys can be found in the textbooks of Lindemann et al. (2009) or Eppinger and Browning (2012). Browning (2001) distinguishes between two basic types of DSMs: static and time-based. Static DSMs represent either product components or teams in an

organization that exist simultaneously. Time-based DSMs either represent dynamic activities indicating precedence relationships or design parameters that change as a function of time. Generated static DSMs are usually analyzed for structural characteristics or by clustering algorithms (e.g. Rogers et al. 2006), whereas time-based DSMs are typically used to optimize workflows based on sequencing, tearing and banding algorithms (e.g. Gebala and Eppinger 1991; Maurer 2007). Kreimeyer and Lindemann (2011) review and discuss a comprehensive set of metrics that can be applied to assess the structure of engineering design processes encoded by DSMs (and other forms). According to Browning's taxonomy, the WTM as dynamical operator of state equation 8 is a static task-based DSM, because the development tasks are processed concurrently and persistent feedback/feed forward loops are modeled through the off-diagonal elements. The majority of work on complexity management with static DSMs focuses on the concept of modularity in identifying cluster structures (see Baldwin and Clark 2000). This work has been very influential in academia and industry. An important limitation, however, is its purely static view of the product structure and, consequently, of the task structure and the interactions between them. A task processing on different time scales corresponding to different autonomous task processing rates cannot be represented. Recent publications indicate that technical dependencies in product families tend to be volatile and therefore coordination needs among development tasks can evolve over time (e.g. Cataldo et al. 2006, 2008; Sosa 2008). When these evolving coordination needs are not adequately managed, significant misalignments of organizational structure and product architecture can occur that have a negative effect on product quality (Gokpinar et al. 2010). An effective method for dealing with volatility of dependencies is to use different WTMs for different phases of the project in which no task is theoretically processed independently of the others. Furthermore, additional task-mapping matrices can be specified at the transition points between phases. By doing so, the number of tasks as well as the kind and intensity of coordination needs can be adapted. It is also possible to specify phase-dependent covariances of performance fluctuations. In many PD projects the performance fluctuations tend to be larger for late development stages that are close to the desired start of production. Another limitation of the concept of product modularity is that the organizational patterns of a development project (e.g. communication links, team co-membership) do not necessarily mirror the technical dependency structures (Sosa et al. 2004). The literature review by Colfer and Baldwin (2010) shows that the "mirroring hypothesis" was supported in only 69% of the cases. Support for the hypothesis was strongest in the within-firm sample, less strong in the across-firm sample, and relatively weak in the open collaborative sample. As such, WTMs and covariance matrices represent dynamic dependency structures in their own right. They must be related to product components or organizational elements through additional multiple domain mapping matrices (Danilovic and Browning 2007) and cannot be substituted by the traditional modeling elements.

An approach to measuring structural complexity based on static component-based DSMs that is formally similar to our own analysis in the spectral basis (see Sections 2.3 and 4.2) has recently been developed by Sinha and de Weck

(2011; 2012). The three terms of their metric C_{SW} are related to the complexities of each of the n components in the system (local effect, represented by the α_i's), the number and complexity of each pairwise interaction (local effect, represented by the β_{ij}'s and a_{ij}'s) and the arrangement of the m interfaces (global, system level effect, represented by $E(A)$). Moreover, a normalization factor γ is introduced. The definition is (Denman et al. 2011; Sinha and de Weck 2012; Sinha 2014):

$$C_{SW} := \sum_{i=1}^{n} \alpha_i + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} a_{ij} \right) \gamma E(A).$$

The normalization factor γ is taken as $1/n$ and used to map the n different components in the system onto a comparable scale. The matrix A is an adjacency matrix that corresponds to the component-based DSM of the product as follows:

$$A = (a_{ij}) = \begin{cases} 1 & \forall (i,j) : (i \neq j) \wedge (i,j) \in \Upsilon \\ 0 & otherwise. \end{cases}$$

The exogenous variable Υ represents the set of connected nodes in the system. Accordingly, the adjacency matrix is simply a binary form of the component-based DSM, in which ones are placed in the cells with marks and zeros elsewhere. The diagonal elements of A are zero. The underlying concept of the metric C_{SW} is that in order to develop the individual components, a non-zero complexity is involved. This complexity can vary across components and is represented by the α_i's, the so-called component complexity estimate (Sinha and de Weck 2012; Sinha 2014). Similar arguments hold true for the complexity β_{ij} of each interface, the so-called final interface complexity (Sinha and de Weck 2012; Sinha 2014). If there are multiple types of interface between two components (energy flow, material flow, control action flow etc.), large beta coefficients are assigned, since it would require more effort to implement them compared to a simpler (univariate) connection. An important aspect is that the correlation between the component complexity estimate and the final interface complexity can vary depending on the kind of product. For large-scale mechanical systems, the β_{ij}'s are often much smaller than the α_i's and α_j's. However, in micro or nanoscale systems it can be the opposite, because it is often much more difficult to develop the interfaces (Sinha 2014). The different interface complexities can be captured using a multiplicative model

$$\beta_{ij} = f_{ij} \alpha_i \alpha_j,$$

where f_{ij} stands for the interface complexity factor (Eppinger and Browning 2012; Sinha and de Weck 2012; Sinha 2014). Finally, the term $E(A)$ represents the graph energy of the adjacency matrix A. The graph energy is defined as the sum of the singular values σ_i of the orthogonal vectors:

$$E(A) := \sum_{i=1}^{n} \sigma_i,$$

where the singular values are computed by the decomposition

$$A = U \cdot \Sigma_A \cdot V^{\mathrm{T}}$$
$$\Sigma_A = \mathrm{Diag}[\sigma_i].$$

The graph energy is invariant under isomorphic transformations (Weyuker 1988) and therefore highly objective.

Ameri and Summers (Ameri et al. 2008; Summers and Ameri 2008) developed a complementary connectedness measure and an algorithm for assessing design connectivity complexity based on graphical models. In the graphical models, the development tasks are nodes of a graph and connected through variable dependency. The algorithm manipulates the graph in terms of connectivity. This manipulation starts by eliminating all unary relations, as they do not contribute to the connectivity complexity of the graph. Once the unary relations have been removed, the score keeping variables are initialized. From this point forward, the graph connectivity algorithm is a recursive algorithm that is applied against all subgraphs that are generated in the process. A cumulative score is maintained to quantify the connectedness of the whole structure (see Summers and Shah 2010). This approach also seems to have certain limitations for assessing emergent complexity in PD projects. The graph of development tasks is recursively decomposed into subgraphs, which tears apart potentially important indirect connections that can lead to higher-order interactions between activities. Furthermore, due to the deterministic approach to modeling the work processes it is impossible to analyze or evaluate the "problem-solving oscillations" (Mihm et al. 2003; Mihm and Loch 2006) emerging from cooperative task processing in conjunction with performance variability. Consequently, we will not consider the design connectivity complexity in the following.

The interested reader can find additional approaches to measuring and evaluating complexity in engineering design with a specific focus on structural characteristics in the excellent textbook by Kreimeyer and Lindemann (2011). The authors present a total of 52 complexity metrics from different disciplines and show in three case studies from process management in the automotive industry how different facets of complexity can materialize in real design processes. They also introduce the Structural Goal Question Metric framework for selecting metrics in a goal-oriented manner and guiding their application.

The information-theory and dependency-structure-based complexity metrics from theories of systematic engineering design are undoubtedly beneficial in facilitating studies that require the use of equivalent but different design problems and in comparing computer-aided design automation tools. Nevertheless, in the following analytical Chapter 4 we will shift our focus to the EMC metric first put forward in Grassberger's seminal theoretical work (1986), as it can both effectively

measure self-generated complexity and provide a foundation for deriving closed-form solutions of different strengths from first principles. Furthermore, EMC stresses the dynamic nature of cooperative work in PD projects and can be calculated efficiently from generative models or from historical data.

Also very interesting for applications in project management is the later-formulated persistent mutual information EMC(τ) (Section 3.2.4). This is partly because of its intimate relationship with the famous Lyapunov function (Nicolis and Nicolis 2007) of a process, and partly because the generated complexity "landscape" often becomes more and more informative as the lead time increases. However, this phenomenon goes beyond the scope of this book and will be analyzed in detail in future work. To lay the analytical foundations for future studies of emergent complexity we will present closed-form solutions of the persistent mutual information for the developed vector autoregression models in the corresponding chapters. These solutions are generalized from the expressions for $\mathrm{EMC} = \mathrm{EMC}(\tau = 0)$, which will be presented in the beginning of Sections 4.1.1, 4.1.2 and 4.1.3 (see Eqs. 247, 253, 262 and 265). Due to the limited space in this book, the closed-form solutions of the persistent mutual information that is generated by a linear dynamical system (Section 2.9) will not be presented. The interested reader can develop them by applying the solution principles that will be introduced in Section 4.2.

The purely information-theoretic view on emergent complexity also opens EMC and the corresponding persistent mutual information EMC(τ) to criticism. In their latest paper on effective complexity (see also Section 3.2.4) Gell-Mann and Lloyd (2004) point out that, without modification, EMC assigns two identical and very long bit strings consisting entirely of 1's with high complexity values because the mutual information between them is very large, yet each process representation is obviously very simple. This is in stark contrast to the fundamental ideas of their EC metric (Eq. 209), which evaluates the algorithmic information content of the strings. The ideal ensemble for modeling an identical very long bit string x is the Dirac measure δ_x, i.e. the ensemble with $\delta_x(x) = 1$ and $\delta_x(x') = 0$ for $x \neq x'$. This ensemble has Kolmogorov−Chaitin complexity $K_U[\delta_x(x)] = K_U(x)$ and information entropy $H[\mathbb{E}] = 0$ (Ay et al. 2010). Its total information $\Sigma[\mathbb{E}]$ is therefore minimal. The algorithmic complexity $K_U(x)$ is apparently very low because the computer program used to calculate x on U simply outputs $|x|$ 1's in a simple pre- or post-test loop. Shiner et al. (2000) also criticize the fact that EMC is not uniquely defined for higher dimensional systems, e.g. spins in two dimensions. In spite of these apparent conceptual weaknesses, the ability of both measures to quantify the degree of informational structure between past and future histories of cooperative task processing and the value of that information in helping to make predictions mean that they are especially interesting and valuable for analyzing, evaluating and optimizing PD projects.

More details on complexity measures from statistical physics, information theory and computer science are presented in Shalizi (2006), Prokopenko et al. (2009), Nicolis and Nicolis (2007), Ellison et al. (2009) and Crutchfield et al. (2010). A focused review of complexity measures for the evaluation of

human−computer interaction including two empirical validation studies can be found in Schlick et al. (2006, 2010).

References

Abdallah, S.A., Plumbley, M.D.: A measure of statistical complexity based on predictive information with application to finite spin systems. Phys. Lett. A **376**, 275–281 (2012)

Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Petrov, B.N., Csaki, F. (eds.) Second International Symposium of Information Theory, pp. 267–281. Akademia Kiado, Budapest, Budapest (1973)

Akaike, H.: A new look at the statistical model identification. IEEE Trans. Automatic Cont **19**, 716–723 (1974)

Amaral, L.A.N., Uzzi, B.: Complex systems: A new paradigm for the integrative study of management, physical, and technological systems. Manag. Sci **53**(7), 1033–1035 (2007)

Ameri, F., Summers, J.D., Mocko, G.M., Porter, M.: Engineering design complexity: An experimental study of methods and measures. Res. Eng. Des. **19**(2-3), 161–179 (2008)

Arnold, D.: Information-theoretic analysis of phase transitions. Complex Syst. **10**(2), 143–155 (1996)

Ay, N., Müller, M., Szkola, A.: Effective complexity and its relation to logical depth. IEEE Trans. Inform. Theor. **56**(9), 4593–4607 (2010)

Ay, N., Bernigau, H., Der, R., Prokopenko, M.: Information driven self-organization: The dynamic system approach to autonomous robot behavior. Theor. Biosci. **131**, 161–179 (2012)

Baldwin, C.Y., Clark, K.B.: Design Rules: The Power of Modularity. MIT Press, Cambridge, MA (2000)

Ball, R.C., Diakonova, M., MacKay, R.S.: Quantifying emergence in terms of persistent mutual information. (2010) arXiv:1003.3028v2 [nlin.AO]

Bennett, C.: Logical depth and physical complexity. In: Herken, R. (ed.) The Universal Turing Machine—a Half-Century Survey, pp. 227–257. Oxford University Press, Oxford (1988)

Bialek, W., Nemenman, I., Tishby, N.: Predictability, complexity and learning. Neural Comput. **13** (1), 2409–2463 (2001)

Bosch-Rekveldt, M., Jongkind, Y., Mooi, H., Bakker, H., Verbraeck, A.: Grasping project complexity in large engineering projects: The TOE (Technical, organizational and environmental) framework. Int. J. Project Manag. **29**(6), 728–739 (2011)

Braha, D., Bar-Yam, Y.: The statistical mechanics of complex product development: Empirical and analytical results. Manag. Sci. **53**(7), 1127–1145 (2007)

Braha, D., Maimon, O.: The measurement of a design structural and functional complexity. IEEE Trans. Syst. Man Cybernet. Part A: Syst. Hum **28**(4), 527–535 (1998)

Browning, T.: Applying the design structure matrix to system decomposition and integration problems: A review and new directions. IEEE Trans. Eng. Manag. **48**(3), 292–306 (2001)

Burnham, K.P., Anderson, D.R.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer, New York, NY (2002)

Carlile, P.R.: A pragmatic view of knowledge and boundaries: Boundary objects in new product development. Organ. Sci. **13**(4), 442–455 (2002)

Cataldo, M., Wagstrom, P.A., Herbsleb, J.D., Carley, K.M.: Identification of Coordination requirements: Implications for the design of collaboration and awareness tools. In: Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work, CSCW 2006, Banff, Alberta, Canada, pp. 353–362, (2006)

Cataldo, M., Herbsleb, J.D., Carley, K.M.: Socio-technical congruence: A framework for assessing the impact of technical and work dependencies on software development productivity. In:

Proceedings of the 2nd International Symposium on Empirical Software Engineering and Measurement (ESEM'08), Kaiserslautern, Germany. pp. 2–11, (2008)
Chaitin, G.J.: Algorithmic Information Theory. Cambridge University Press, Cambridge (1987)
Colfer, L.J., Baldwin, C.Y.: The mirroring hypothesis: Theory, evidence and exceptions. Harvard Business School Working Paper 10-058, (2010)
Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley and Sons, New York (1991)
Crutchfield, J.P., Feldman, D.P.: Regularities unseen, randomness observed: Levels of entropy convergence. Chaos **13**(25), 25–54 (2003)
Crutchfield, J.P., Young, K.: Inferring statistical complexity. Phys. Rev. Lett. **63**, 105–108 (1989)
Crutchfield, J.P., Young, K.: Computation at the onset of Chaos. In: Zurek, W.H. (ed.) Complexity, entropy, and the physics of information, pp. 223–269. Addison-Wesley, Reading, MA (1990)
Crutchfield, J.P., Ellison, C.J., James, R.G., Mahoney, J.R.: Synchronization and control in intrinsic and designed computation: an information-theoretic analysis of competing models of stochastic computation. Santa Fe Institute Working Paper 2010-08-015. (2010)
Crutchfield, J.P., Marzen, S.: Signatures of infinity: Nonergodicity and resource scaling in prediction, complexity, and learning. Santa Fe Institute Working Paper 2015-04-010. (2015)
Cummings, J.N., Espinosa, J.A., Pickering, C.K.: Crossing spatial and temporal boundaries in globally distributed projects: A relational model of coordination delay. Inform. Syst. Res. **20** (3), 420–439 (2009)
Danilovic, M., Browning, T.R.: Managing complex product development projects with design structure matrices and domain mapping matrices. Int. J. Project Manag. **25**(3), 300–314 (2007)
deLeeuw, J.: Introduction to Akaike (1973) information theory and an extension of the maximum likelihood principle. In: Kotz, S., Johnson, N.L. (eds.) Breakthroughs in Statistics, vol. 1, pp. 599–609. Springer, London (1992)
Denman, J., Kaushik, S., de Weck, O.: Technology insertion in Turbofan Engine and assessment of architectural complexity. In: Proceedings of the 13th International Dependency and Structure Modeling Conference, DSM 2011, pp. 407–420, (2011).
Dvir, D., Sadeh, A., Malach-Pines, A.: Projects and Project Managers: The relationship between projects managers' personality, project types, and project success. Project Manag. J. **37**(5), 36–48 (2006)
Edwards, A.W.F.: Likelihood. Cambridge University Press, Cambridge U.K. (1972)
El-Haik, B., Yang, K.: The components of complexity in engineering design. IIE Trans. **31**(10), 925–934 (1999)
Ellison, C.J., Mahoney, J.R., Crutchfield, J.P.: Prediction, retrodiction, and the amount of information stored in the present. Santa Fe Institute Working Paper 2009-05-017. (2009)
Eppinger, S.D., Browning, T.: Design Structure Matrix Methods and Applications. MIT Press, Cambridge, MA (2012)
Foley, D.K., Oliver, D.: Notes on Bayesian inference and effective complexity. Unpublished manuscript. Available at http://www.american.edu/cas/economics/info-metrics/pdf/upload/Oct-2011-Workshop-Paper-Foley-and-Oliver.pdf (2011) (retrieved September 2013).
Gebala, D.A., Eppinger, S.D.: Methods for analyzing design procedures. In: Proceedings of the ASME Conference on Design Theory and Methodology, Miami, FL, pp. 227–233, (1991)
Gell-Mann, M.: What is complexity. Complexity **1**(1), 16–19 (1995)
Gell-Mann, M., Lloyd, S.: Information measures, effective complexity, and total information. Complexity **2**(1), 44–52 (1996)
Gell-Mann, M., Lloyd, S.: Effective complexity. In: Gell-Mann, M., Tsallis, C. (eds.) Nonextensive Entropy—Interdisciplinary Applications, pp. 387–398. Oxford University Press, Oxford (2004)
Gharahmani, Z.: An introduction to hidden Markov Models and Bayesian Networks. Int. J. Pattern Recogn. Artif. Intell. **15**(1), 9–42 (2001)

Gokpinar, B., Hopp, W.J., Iravani, S.M.R.: The impact of misalignment of organizational structure and product architecture on quality in complex product development. Manag. Sci. **56**(3), 468–484 (2010)
Grassberger, P.: Toward a quantitative theory of self-generated complexity. Int. J. Theor. Phys. **25** (9), 907–938 (1986)
Griffin, A.: The effect of project and process characteristics on product development cycle time. J. Market. Res. **34**(1), 24–35 (1997)
Grünwald, P.: The minimum description length principle. MIT Press, Cambridge, MA (2007)
Gu, M., Wiesner, K., Rieper, E., Vedral, V.: Quantum mechanics can reduce the complexity of classical models. Nat. Commun. **3**, Article number: 762. (2012)
Hansen, M.H., Yu, B.: Model selection and the principle of minimum description length. J. Am Stat. Soc. **96**(454), 746–774 (2001)
Hass, K.B.: Managing Complex Projects. A New Model. Management Concepts. Leesburg Pike, PA (2009)
Hölttä-Otto, K., Magee, C.L.: Estimating factors affecting project task size in product development—An empirical study. IEEE Trans. Eng. Manag. **53**(1), 86–94 (2006)
James, R., Ellison, C.J., Crutchfield, J.P.: Anatomy of a bit: Information in a time series observation. Santa Fe Institute Working Paper 2011-05-019. (2011)
Jänicke, H., Scheuermann, G.: Steady visualization of the dynamics in fluids using ε-machines. Comput. Graph. **33**(1), 597–606 (2009)
Kellogg, K.C., Orlikowski, W.J., Yates, J.: Life in the trading zone: Structuring coordination across boundaries in postbureaucratic organizations. Organ. Sci. **17**(1), 22–44 (2006)
Kerzner, H.: Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons, Hoboken, NJ (2009)
Kim, J., Wilemon, D.: Sources and assessment of complexity in NPD projects. R&D Manag. **33** (1), 15–30 (2003)
Kim, J., Wilemon, D.: An empirical investigation of complexity and its management in new product development. Technol. Analysis Strat. Manag. **21**(4), 547–564 (2009)
Koppel, M., Atlan, H.: An almost machine-independent theory of program-length complexity, sophistication, and induction. Inform. Sci. **56**(1-3), 23–33 (1991)
Kreimeyer, M., Lindemann, U.: Complexity Metrics in Engineering Design—Managing the Structure of Design Processes. Springer, Berlin (2011)
Ladyman, J., Lambert, J., Wiesner, K.: What is a complex system? Eur. J. Philos. Sci. **3**(1), 33–67 (2013)
Lebcir, M.R.: Impact of Project Complexity Factors on New Product Development Cycle Time. University of Hertfordshire Business School Working Paper. https://uhra.herts.ac.uk/dspace/handle/2299/5549. (2011)
Li, W.: On the relationship between complexity and entropy for Markov chains and regular languages. Complex Syst. **5**(4), 381–399 (1991)
Li, L.: Some notes on mutual information between past and future. J. Time Ser. Anal. **27**(2), 309–322 (2006)
Li, M., Vitányi, P.: An introduction to Kolmogorov complexity and its applications, 2nd edn. Springer, New York, NY (1997)
Li, L., Xie, Z.: Model selection and order determination for time series by information between the past and the future. J Time Ser. Anal **17**(1), 65–84 (1996)
Lind, M., Marcus, B.: An Introduction to Symbolic Dynamics and Coding. Cambridge University Press, Cambridge (1995)
Lindemann, U., Maurer, M., Braun, T.: Structural Complexity Management. An Approach for the Field of Product Design. Springer, Berlin (2009)
Löhr, W.: Predictive models and generative complexity. J. Syst. Sci. Complex. **25**(1), 30–45 (2012)
MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge U.K. (2003)

Marzen, S., Crutchfield, J.P.: Circumventing the curse of dimensionality in prediction: Causal rate-distortion for infinite-order markov processes. Santa Fe Institute Working Paper 2014-12-047, (2014)
Maurer, M.: Structural awareness in complex product design. Doctoral dissertation, Technische Universität München. Dr. Hut Verlag, Munich, Germany, (2007)
Maylor, H., Vidgen, R., Carver, S.: Managerial complexity in project-based operations: A grounded model and its implications for practice. Project Manag. J. **39**(1), 15–26 (2008)
Mihm, J., Loch, C., Huchzermeier, A.: Problem-solving oscillations in complex engineering. Manag. Sci. **46**(6), 733–750 (2003)
Mihm, J., Loch, C.: Spiraling out of control: Problem-solving dynamics in complex distributed engineering projects. In: Braha, D., Minai, A.A., Bar-Yam, Y. (eds.) Complex Engineered Systems: Science Meets Technology, pp. 141–158. Springer, Berlin (2006)
Mihm, J., Loch, C., Wilkinson, D., Huberman, B.: Hierarchical structure and search in complex organisations. Manag. Sci. **56**(5), 831–848 (2010)
Mulenburg, J.: What does complexity have to do with it? Complexity and the management of projects. In: Proceedings of the 2008 NASA Project Management Challenge Conference. (2008)
Murmann, P.A.: Expected development time reductions in the german mechanical engineering industry. J. Product Innov. Manag. **11**(3), 236–252 (1994)
Nicolis, G., Nicolis, C.: Foundations of Complex Systems: Nonlinear Dynamics, Statistical Physics, Information and Prediction. World Scientific, Singapore (2007)
O'Leary, M.B., Mortensen, M.: Go (Con)figure: Subgroups, imbalance, and isolates in Geographically dispersed teams. Organ. Sci. **21**(1), 115–131 (2010)
Papoulis, A., Pillai, S.U.: Probability, random variables and stochastic processes. McGraw-Hill, Boston, MA (2002)
Polani, D., Nehaniv, C., Martinetz, T., Kim, J.T.: Relevant information in optimized persistence vs. progeny strategies. In: Proceedings of The 10th International Conference on the Simulation and Synthesis of Living Systems, Artificial Life X, pp. 337–343, (2006)
Prokopenko, M., Boschetti, F., Ryan, A.J.: An information-theoretic primer on complexity, self-organization and emergence. Complexity **15**(1), 11–28 (2009)
Puri, N.N.: Fundamentals of linear systems for physical scientists and engineers. CRC Press, Boca Raton, FL (2010)
Rissanen, J.: Stochastic complexity in statistical inquiry. World Scientific, Singapore (1989)
Rissanen, J.: Fisher information and stochastic complexity. IEEE Trans. Inform. Theor. **42**(1), 40–47 (1996)
Rissanen, J.: Information and Complexity in Statistical Modeling. Springer, Berlin (2007)
Rissanen, J.: Optimal Estimation of Parameters. Cambridge University Press, Cambridge (2012)
Rivkin, J.W., Siggelkow, N.: Balancing search and stability: Interdependencies among elements of organizational design. Manag. Sci. **49**(3), 290–311 (2003)
Rivkin, J.W., Siggelkow, N.: Patterned interactions in complex systems: Implications for exploration. Manag. Sci. **53**(7), 1068–1085 (2007)
Rogers, J.L., Korte, J.J., Bilardo, V.J. Development of a genetic algorithm to automate clustering of a dependency structure matrix. National Aeronautics and Space Administration, Langley Research Center, Technical Memorandum NASA/TM-2006-214279, (2006)
Schlick, C.M., Winkelholz, C., Motz, F., Luczak, H.: Self-generated complexity and human–Machine interaction. IEEE Trans. Syst. Man Cybernet, Part A: Syst. Hum **36**(1), 220–232 (2006)
Schlick, C.M., Beutner, E., Duckwitz, S., Licht, T.: A complexity measure for new product development projects. In: Proceedings of the 19th International Engineering Management Conference, pp. 143−150, (2007)
Schlick, C.M., Duckwitz, S., Gärtner, T., Tackenberg, S.: Optimization of concurrent engineering projects using an information-theoretic complexity metric. In: Proceedings of the 11th International DSM Conference, pp. 53–64, (2009)

Schlick, C.M., Winkelholz, C., Motz, F., Duckwitz, S., Grandt, M.: Complexity assessment of human–Computer interaction. Theor. Iss. Ergon. Sci. **11**(3), 151–173 (2010)
Shalizi, C.R.: Methods and techniques of complex systems science: An overview. In: Deisboeck, T.S., Kresh, J.Y. (eds.) Complex systems science in biomedicine, pp. 33–114. Springer, New York (2006)
Shalizi, C.R., Crutchfield, J.P.: Computational mechanics: Pattern and prediction, structure and simplicity. J. Stat. Phys. **104**, 817–879 (2001)
Shalizi, C.R., Shalizi, K.L.: Optimal nonlinear prediction of random fields on networks. In: Discrete Mathematics and Theoretical Computer Science, AB(DMCS), pp. 11–30, (2003)
Shalizi, C.R., Shalizi, K.L.: Blind construction of optimal nonlinear recursive predictors for discrete sequences. In: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence, pp. 504–511, (2004)
Shaw, R.: The Dripping Faucet as a Model Chaotic System. Aerial Press, Santa Cruz, CA (1984)
Shenhar, A.J.: From theory to practice: Toward a typology of project management styles. IEEE Trans. Eng. Manag. **45**(1), 33–48 (1998)
Shenhar, A.J., Dvir, D.: Toward a typological theory of project management. Res. Pol. **25**(4), 607–632 (1996)
Shenhar, A.J., Dvir, D.: Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation. Harvard Business School Press, Boston, MA (2007)
Shiner, J.S., Davison, M., Landsberg, P.T.: Reply to comments on "Simple measure for complexity". Phys. Rev. E **62**(2), 3000–3003 (2000)
Shtarkov, Y.M.: Universal sequential coding of single messages. Prob. Inform. Transmis. **23**(3), 3–17 (1987) (translated from Russian)
Shtub, A., Bard, J.F., Globerson, S.: Project Management—Processes, Methodologies, and Economics, 2nd edn. Prentice Hall, Upper Saddle River, NJ (2004)
Sinha, K., de Weck, O.: Spectral and topological features of "Real-World" Product Structures. In: Proceedings of the 11th International Dependency and Structure Modeling Conference, DSM 2011, pp. 65–77, (2011).
Sinha, K., de Weck, O.: Structural complexity metric for engineered complex systems and its application. In: Proceedings of the 12th International Dependency and Structure Modeling Conference, DSM 2012, pp. 181–192, (2012)
Sinha, K.: Structural Complexity and its Implications for Design of Cyber-Physical Systems. Ph. D. Thesis, Massachusetts Institute of Technology, (2014)
Smith, R.P., Eppinger, S.D.: Identifying controlling features of engineering design iteration. Manag. Sci. **43**(3), 276–293 (1997)
Sosa, M.E.: A structured approach to predicting and managing technical interactions in software development. Res. Eng. Des. **19**, 47–70 (2008)
Sosa, M.E., Eppinger, S.D., Rowles, C.M.: The misalignment of product architecture and organizational structure in complex product development. Manag. Sci. **50**(12), 1674–1689 (2004)
Steward, D.V.: The design structure system: A method for managing the design of complex systems. IEEE Trans. Eng. Manag. **28**(3), 71–74 (1981)
Suh, N.P.: Axiomatic Design: Advances and Applications. Oxford University Press, Oxford (2001)
Suh, N.P.: Complexity—Theory and Applications. Oxford University Press, Oxford (2005)
Summers, J.D., Shah, J.J.: Mechanical engineering design complexity metrics: Size, coupling, and solvability. J. Mech Des. **132**(2), 1–11 (2010)
Summers, J.D., Shah, J.J.: Developing measures of complexity for engineering design. In: Proc. ASME DETC, Chicago, IL, Paper DTM-48633. pp. 381–392, (2003)
Summers, J.D., Ameri, F.: An algorithm for assessing design connectivity complexity. In: Tools and Methods for Competitive Engineering Conference, Izmir, Turkey, (2008)
Tatikonda, M.V., Rosenthal, S.R.: Technology novelty, project complexity and product development project execution success. IEEE Trans. Eng. Manag. **47**, 74–87 (2000)

Travers, N.F., Crutchfield, J.P.: Infinite excess entropy processes with countable-state generators. Entropy **16**(3), 1396–1413 (2014)
Travers, N.F., Crutchfield, J.P.: Equivalence of History and generator epsilon-machines. Santa Fe Institute Working Paper 2011-11-015, (2011).
Vitányi, P., Li, M.: Minimum description length induction, Bayesianism, and Kolmogorov Complexity. IEEE Trans. Inform. Theor **46**(2), 446–464 (2000)
Wallace, C.S., Boulton, D.M.: An information measure for classification. Comput J. **11**(2), 185–195 (1968)
Weyuker, E.: Evaluating software complexity measures. IEEE Trans. Softw. Eng. **14**(9), 1357–1365 (1988)
Wheelwright, S.C., Clark, K.B.: Creating project plans to focus product development. Harv. Bus. Rev. **70**(2), 70–82 (1992)
Wiesner, K.: Complexity measures and physical principles. In: Sanayei, A. et al. (eds). ISCS 2014: Interdisciplinary Symposium on Complex Systems, Emergence, Complexity and Computation, vol. 14, pp 15–20, (2015)
Zambella, D., Grassberger, P.: Complexity of forecasting in a class of simple models. Complex Syst. **2**(1), 269–303 (1988)