System of Systems Characteristics in Production Systems Engineering

Marcus Bjelkemyr

Doctoral Thesis in Production Engineering
Stockholm, Sweden 2009
To Maria and Jack

Without obsession,
life is nothing.
- John Waters
Abstract

This thesis presents a systems view of production, where production systems are compared and contrasted with other large and complex systems, commonly labeled System of Systems (SoS). The rationale for this approach lies in the evolution of production systems towards being holistic, sustainable, and agile; which increases the need for an improved understanding of both how internal system are interrelated, and how the production system interacts with its environment. In turn, this leads to an increase of complexity for the production system, which leads to new requirements on systems engineering.

The definition of SoS is extensively discussed, and in this thesis formalized with regards to certain system characteristics that SoS exhibit. The presence of these characteristics is evaluated for three different levels of production systems to determine if they should be considered SoS. In the second part of the thesis, the SoS characteristics are addressed from an engineering point of view, i.e. if and how SoS properties are currently addressed in production systems engineering.

Two main results are presented in this thesis: (1) production systems exhibit SoS characteristics; (2) SoS characteristics are not and cannot be addressed with current systems engineering methods. How SoS characteristics can be addressed is briefly discussed in the frame of reference.

An additional purpose of this thesis is to initiate a new research area where production systems research and complex systems research are merged.
Acknowledgements

I would like to express my gratitude to all the people who gave me the possibility to complete this thesis. First of all I would like to thank my supervisor Professor Bengt Lindberg for his guidance and contributions throughout my work. His open-mindedness has allowed me to explore production research into new directions. He has always been a great support, both as my supervisor and my boss.

Professor emeritus Gunnar Sohlenius has provided great scientific inspiration and the initial contact with Professor Nam P. Suh, my supervisor at Massachusetts Institute of Technology. At MIT I would also like to thank Professor Magee and Professor Cutcher-Gershenfield for inspiring my scientific turn towards complex systems.

Returning to KTH and the Department for Production Engineering I would like to thank Dr. Dario Aganovic and Jonas Fagerström for pulling me in to the department, for initially being my unofficial secondary supervisors, for inspiration, and for being wonderful colleagues and friends.

In my research group of Evolvable Production Systems I would particularly like to thank Associate Professor Mauro Onori and Dr. Daniel Semere for the scientific discussions and for the rigorous review of this thesis.

I would also like to thank my other colleagues over the years, both at the department at KTH and at the Park Center for Complex Systems at MIT.

My financers should not be forgotten: Woxencenter, VINNOVA through the MERA Programme, and the Hans Werthén Fund for enabling my stay at MIT.

Finally I would like to thank my family and friends, in particular my wife Maria for providing me with precious time and my son Jack for not doing so.
Publications

The work presented in this thesis is based on research detailed in the previously published papers below. Papers 1-5 are of particular relevance for the thesis and are therefore appended.


Additional publications


Table of Contents

Abstract ...........................................................................................................................................i
Acknowledgements ..................................................................................................................ii
Publications ..................................................................................................................................iii
Table of Contents ....................................................................................................................v
List of Figures ............................................................................................................................vii
List of Tables .............................................................................................................................viii

1 Introduction .............................................................................................................................1
  1.1 Scope of Thesis ..................................................................................................................2
  1.2 Research Methodology .................................................................................................3
  1.3 Research Question and Hypotheses ............................................................................4
  1.4 Limitations ......................................................................................................................7
  1.5 Outline of Thesis ...........................................................................................................8

2 Frame of Reference .................................................................................................................9
  2.1 System Classification ......................................................................................................9
    2.1.1 Definition of System and Environment ..............................................................10
    2.1.2 Definition of Production System ........................................................................13
    2.1.3 Definition of System of Systems .......................................................................17
    2.1.4 Characteristics of System of Systems ..............................................................21
  2.2 Systems engineering .......................................................................................................27
    2.2.1 Reducible Systems Engineering ........................................................................27
    2.2.2 Systems Engineering Methods ..........................................................................29
    2.2.3 RS and SoS Differences, and Effect on SoSE ....................................................32
    2.2.4 Decision Making in SoSE ..................................................................................33
    2.2.5 A Biological Approach to SoSE .......................................................................35
  2.3 Conceptualization Framework .........................................................................................37
    2.3.1 The Substance of a Conceptualization ...............................................................39
    2.3.2 Limitations to Conceptualizing .........................................................................41

3 Results .....................................................................................................................................43
  3.1 Hypothesis A ....................................................................................................................43
    3.1.1 SoS Characteristics in Production Systems .......................................................43
    3.1.2 Discussion of Hypothesis A ...............................................................................55
3.2 Hypothesis B ........................................................................................................57
3.2.1 Effects of the Limitations to Conceptualizing .....................................57
3.2.2 Discussion of Hypothesis B .................................................................63
4 Final Discussion and Conclusion .................................................................67
4.1 Critical Review ............................................................................................67
4.2 Future Research ..........................................................................................69
4.3 Conclusions ..................................................................................................70
References ..........................................................................................................71
List of Figures

Figure 1: Illustration of the relationship between the research question and the two hypotheses. .................................................................5

Figure 2: Illustration of how different thesis chapters are related to each other and to the appended publications. .........................................................8

Figure 3: Illustration of how the chapters in the 'Frame of Reference' are related to the general feature of Complex Systems and Reducible Systems. ..........9

Figure 4: Illustration of the definition of a system and its environment, understood through the idea that each and every possible pattern either belongs to the environment or the system. ........................................11

Figure 5: Relationship between the terms Manufacturing System, Production System, and Extended Production System. ...........................................14

Figure 6: The methodology scheme for an Evolvable Assembly System (EAS), illustrates how an evolvable system is able to adapt to change on a module level (Maffei et al. 2009). .................................................................16

Figure 7: Concept model of the definition for a SoS. .................................................21

Figure 8: Small-world in comparison to regular and random models of networks (Watts and Strogatz 1998). .................................................................24

Figure 9: Comparison between an exponential and a scale-free network (Albert, Jeong, and Barabasi 2000). .................................................................25

Figure 10: A power law distribution; y-axis denotes the nodal degree, x-axis denotes the number of nodes with a particular nodal degree. .................26

Figure 11: System life cycle processes of ISO/IEC 15288, divided into four key areas: enterprise, agreement, project, and technical processes (ISO 2002). .29

Figure 12: Illustration of the high-level processes in the ModArt aid for model driven engineering of a manufacturing system (ModArt 2009). ........30

Figure 13: Illustration of the interrelations of the key engineering design methods that are the theoretic backbone of KTH-IPM (Dario Aganovic, Marcus Bjelkemyr, and Bengt Lindberg 2003). ........................................31

Figure 14: The Cynefin Framework describes that the appropriate response to a situation is related to the engineering context, i.e. simple, complicated, complex or chaotic (Snowden and Boone 2007). .........................................................33

Figure 15: Concept model of the substance of a conceptualization ..................38

Figure 16: Illustration of how the correlation between field of view and resolution affect our ability to conceptualize everything at once. .................39

Figure 17: Illustration of how resolution and field of view may change which patterns are visible for the observer (M. Bjelkemyr and B. Lindberg 2007). .40

Figure 18: Bi-stable figure of Rabbit/Duck (Jastrow 1899) ...............................41
Figure 19: Automotive suppliers in the Gothenburg area (detail from (Djurberg and Backlund 2008)). .................................................................45

Figure 20: Relation between the stability of a system's processes and the obtainable autonomy (Maffei et al. 2009). ..............................................................46

Figure 21: Design and integration of a manufacturing system for a new product, comparison between a non-modular centralized system and an evolvable production system (Neves and Onori 2009). .........................................................50

Figure 22: Global structure of Ohta supplier-prime buyer network (Nakano and D.R. White 2006). ......................................................................................................52

Figure 23: A system must be decomposed in accordance with an individual's ability to handle information. ...................................................................................58

Figure 24: A system must be decomposed in accordance with an individual's ability to handle only one scale at the same time. ..................................................58

Figure 25: Decomposition of a reducible system. .................................................65

Figure 26: Evolution of a SoS; starts with an intention, is initiated locally and grows in accordance with internal and external relationships..............................65

List of Tables

Table 1: Definition and objective of five manufacturing system types, divided into the categories centralized and distributed systems. ................................................15

Table 2: Example of network measures (excerpt from Table 6 ) .........................26

Table 3: Comparison between SoS and RS, adapted from (M. Bjelkemyr, Semere, and B. Lindberg 2009) and (Norman and M.L. Kuras 2004), and the effect on SoSE. ............................................................................................................32

Table 4: Characteristics of the four engineering contexts and main tasks of the leader, extracted from (Snowden and Boone 2007) ..............................................34

Table 5: Traits of biological development and its relevance to SoSE. .....................36

Table 6: Network measures for product development networks (*) (Braha and Y. Bar-Yam 2004a) and a supply network (†) (Nakano and D.R. White 2006). 54

Table 7: SoS characteristics for different production systems. ............................55

Table 8: Evaluation of correlation between SoS characteristics and the capacity and capability of an individual. ...............................................................63
1 Introduction

Systems engineering has evolved from pen and paper engineering of one single system with a predetermined life-cycle, to model driven concurrent engineering of a multitude of adaptive systems. This transition has been both pulled and pushed: pulled by the current trend towards sustainable and holistic systems, which underlines the need to view production from a more multifaceted perspective; and pulled by technical advancements of mainly computer based engineering tools used both during development and production. Despite this radical change in engineering, the underlying theoretical foundation of current tools and methods are still based on traditional hierarchical systems engineering. This leads to that complex system characteristics are still difficult to address, e.g. self-organization, evolutionary behavior, heterogeneity, emergent behavior, and network properties.

These system properties are have shown to be common properties of many large and complex socio-technical systems, often referred to as complex system, engineering system, socio-technical system, enterprise system, or System of Systems [SoS]; the latter is predominantly used in this thesis. Instances of these systems are present in a variety of areas, but most interest has so far been generated within military and defense, aeronautics, transportation, and infrastructure. This thesis is based around the ideas that production systems are also system of systems, that they consequently exhibit some or all of the above mentioned SoS characteristics, and that traditional systems engineering is insufficient for addressing these system properties.

The following chapters in the introduction are intended to provide the reader with an overview of the thesis and an understanding of how this thesis is positioned academically and industrially: The academic and industrial contribution is discussed; the research question and supporting hypotheses are presented and elaborated; limitations to the research are discussed; and individual chapters in the thesis are related to the hypothesis and to the appended papers in the thesis outline.
1.1 Scope of Thesis

Even though the objects of study are production systems and production systems engineering, the key issues in this thesis are not limited to the area of production. Instead, production systems are considered instances of a category of holistic systems that have both a technical and a societal complexity and that are able to evolve in accordance with their environment. These complex systems will in this thesis be labeled system of systems, and the engineering of SoS is labeled system of systems engineering (SoSE). SoS are functionally and physically vastly dissimilar, but they share similar systemic characteristics, which are studied in an interdisciplinary area of research. Even though the key issues within this area are not limited to a specific engineering field, it is important to maintain the connection to a primary field of engineering in order to obtain case data and to implement results.

In this thesis the overall aim is to verify that production systems are SoS, and to understand how this affects engineering of production systems. By clarifying the link between an individual’s abilities and a system’s characteristics, methods and work processes, the system life cycle can be altered and improved to better suit engineers’ abilities, and thereby improve overall system functionality. The character of this thesis is foundational and its outcome is intended to be used as a basis for improvement of processes, support systems and methods utilized by designers during systems engineering. This can be achieved in both industry and academia; however, the latter is the most likely recipient of the findings presented in this thesis.

The industrial aim is both to show that production systems exhibit SoS characteristics and that limitations of peoples’ abilities affect the engineering of a production system. This knowledge is intended as a basis for comprehending the possibilities and limitations of production systems engineering; especially in understanding that SoSE can aid in minimizing unwanted emergent behavior, improving system fitness, and improving the probability of project success. As stated above, this work is not intended to be directly applied as a manual within industry. It is rather a basis for further work.
The academic contributions are naturally related to answering the research question and the corroboration or falsification of the hypotheses in the thesis. The research question addresses the issue of whether the current engineering practice used for technical systems is able to handle SoS characteristics as well. This research question is addressed through two hypotheses: (A) the first hypothesis establishes a link between different types of production systems and SoS; corroboration of this hypothesis shows that there are, in addition to traditional production system requirements, supplementary system characteristics that have to be considered during production systems engineering; (B) the second hypothesis relates the abilities and limitations of both individuals and engineering methods to the previously defined SoS characteristics; corroboration establishes a human centered approach to production systems engineering, which enables both improvement of existing methods and development of novel method and processes for engineering of SoS.

Another very important result that corroboration of hypothesis A leads to is that a new scientific area within production engineering is established. This can lead to that results from other scientific areas are more easily accessible to production engineers and scientists, and vice versa.

1.2 Research Methodology

The methodological approach in this thesis is based on hypotheses and Popper’s falsification view, i.e. scientific progress is achieved through both falsification and corroboration of hypotheses.

The thesis is based on a research question which I am addressing through two hypotheses. The hypotheses are then further decomposed into predicates which are individually tested. In the first hypothesis, the predicates can be placed in a five by four matrix, totaling 20 predicates; where the columns represent different levels of production systems and the rows represent the components of the definition of SoS. The verification of each predicate is achieved through a process where previously published research that either support or rebut each individual predicate is presented. The result from this approach is a
corroboration or falsification of the hypothesis for four different types of production systems. It is however important to point out that the testing of hypothesis A is fragmented, i.e. the verification has been done for each predicate individually, not for each production system level. This means that the hypothesis has not been verified for a specific production system or production systems from a specific industry.

The second hypothesis is addressed in a similar manner: the rows still represent SoS characteristics, but the columns correspond to limitations of an individual’s ability to engineer a system. The latter is thesis related to the information content that an individual is able to process and the number of scales an individual is able to handle at one time. All ten predicates are then tested logically.

Finally the results from both hypotheses are elevated in order to answer the research question of the thesis.

1.3 Research Question and Hypotheses

This thesis has grown out of a main research question, and the purpose of the thesis is to answer this question:

Does Reducible Systems Engineering sufficiently address all production system characteristics, or is a new approach required?

Reducible Systems Engineering [RSE] is based on reductionism, i.e. “the belief that any portion of reality – including all of it – can be understood or comprehended by understanding the parts of that reality and composing a mental model of the greater reality exclusively from those parts (M. L. Kuras 2007). The term RSE is introduced by Kuras (2007) for what is commonly understood as classical or traditional systems engineering. The reason for using the term RSE is to remove the negative connotation associated with “classical” and “traditional”, and instead focus on when and where it is applicable. Reducible systems engineering is seen as a specified process where the goal is to “produce efficient and reliable systems that meet pre-specified constraints and pre-specified standards of performance in pre-specified situations” (Minai, Braha, and Y. Bar-Yam 2006). The tools, methods and work processes that are used in reducible systems engineering (RSE) are optimized to achieve these goals. Engineering of SoS is
motivated by different goals than RSE; consequently limiting the appropriateness of reducible tools, methods and work processes in the engineering of SoS, and vice versa.

The term *production system* is somewhat recklessly used in the research question. It can and should therefore be interpreted as anything from a single machine to a multinational web of producing companies; later in the thesis four subclasses of production systems are defined and used in the analysis.

![Diagram illustrating the relationship between the research question and the two hypotheses.](image)

Figure 1: Illustration of the relationship between the research question and the two hypotheses.

With this understanding of reductionism, RSE, and production system the research question can be decomposed into two sub questions:

- *Is a production system a reducible or a complex system?*
- *Can RSE handle SoS characteristics?*
To be able to address these questions scientifically, they have been restated in a hypothesis format, which enables testing of their validity and corroborating or falsifying them (Figure 1). The objective is to individually address smaller topics, which can be logically compiled into an answer to the main research question.

**Hypothesis A: Production systems exhibit SoS properties.**

The idea of this hypothesis is to explore to what extent a production system can be considered a system of systems, and thereby not reducible. The rationale lies in understanding which SoS characteristics are at play during production systems engineering, and how each SoS characteristic is constituted in production systems engineering.

Corroboration of hypothesis A establishes a link between production system research and SoS/Complex Systems research; which is an important factor for further research in the area. In addition, corroboration also enables each characteristic to be understood individually; thereby clarifying limitations of current engineering practice and improving the feasibility of gradually incorporating SoS characteristics in production systems engineering.

**Hypothesis B: Limitations to an individual’s capability to conceptualize disable the ability to successfully design SoS with methods for reducible systems engineering.**

The purpose of this hypothesis is to explore if RSE is able to cope with the increasing complexity of sustainable and holistic production systems; driven by a focus on long-term societal, economical, ecological and technological issues. The idea is to increase the understanding of how SoS characteristics emerge and how an individual’s capabilities affect the ability to engineer SoS with reducible SE methods.

Corroboration of hypothesis B enables an individual based understanding of the limitations of reducible systems engineering. These limitations can then be used to develop non-reducible systems engineering methods that are able to answer to the need for sustainable production systems that consider long-term societal, economic, ecological, and technological issues.
1.4 Limitations

The methodological approach in this thesis results in a rather wide scope with regards to types of production system and product segment. In chapter 2.1.2 different types of production system are discussed, defined and classified into three main types: extended production systems, production systems, and manufacturing systems. These three are defined based on which supporting systems are considered as part of the system itself or its environment. Within each of these types, there is a great variance of complexity; however, normally there is a declining degree of complexity between these system types.

It is difficult to make clear demarcation of the validity of the results since the sources used in this thesis cover both a wide range of systems from different industrial areas and ideal representations of systems. The results of the thesis can therefore only be said to be valid for ideal, conceptual manufacturing system concepts and for large and complex sustainable production systems. The results should therefore be seen as indicative, and the validity for any real world production systems must be further analyzed.

In addition it needs to be pointed out that the purpose is not to provide a tool or a method that can be used in the industry to handle SoS issues, much more research is needed to achieve that. The purpose is rather to show how the abilities of an individual determine requirements on tools and methods for engineering of SoS.
1.5 Outline of Thesis

This thesis is based on four main parts, where each part provides a significant contribution to the thesis as a whole (Figure 2). In the first part the thesis is introduced and positioned with regards to academic and industrial relevance, and methodological approach.

In the second part the frame of reference is presented in three sub-chapters: system classification, system engineering, and conceptualization framework. These three chapters form the basis for the research results.

In the third part the research results are presented in the form of two hypotheses, which are related to the research question in accordance with Figure 1.

Finally, a critical review of the thesis is presented together with future research and conclusions.

Figure 2: Illustration of how different thesis chapters are related to each other and to the appended publications.
2 Frame of Reference

This thesis is based on three central areas within systems research: system classification, systems engineering, and an individual’s ability to conceptualize a system (Figure 3).

![Diagram of Complex Systems and Reducible Systems]

Figure 3: Illustration of how the chapters in the 'Frame of Reference' are related to the general feature of Complex Systems and Reducible Systems.

2.1 System Classification

Research on systems engineering has traditionally belonged to industry specific departments, e.g. Aeronautics, Civil and Environmental Engineering, Computer Science, Industrial Production, and Management and Economics. Cross-disciplinary efforts focusing on generic and common system properties have only recently gained interest due to the increasing need to develop sustainable systems that outlive single product generations. Two of the main drivers are to cope with the increasing demand for concurrent collaborations between systems; and the increasing need to develop systems that are sustainable over time that can outlive single product generations. Specific systems departments and institutes have been established, e.g. INCOSE (1995 (NCOSE 1989)), NECSI (1996), MIT’s Engineering Systems Division (1998).
In light of a generic system perspective, challenges facing production systems and production systems engineering are in this thesis not considered unique to the area of production. On the contrary, challenges are often generic and common to a wide variety of systems. To advance production system research more efficiently, non-production methods, tools, and processes developed within other system areas should be studied. However; all systems related research are not of interest for production, which raises the question of how to determine which systems are similar enough to production systems for trans-system exchange to be fruitful. Instead of sorting systems according to area of application, a more useful classification is achieved by organizing them according to characteristics. These could for example be size; complexity; type, e.g. technical, social, or natural; architecture; network properties; openness; et cetera.

A category of systems that is of particular interest for the area of production is system of systems [SoS], which is understood as “a large and complex socio-technical system” (M. Bjelkemyr, Semere, and B. Lindberg 2007). Examples of SoS include but are not limited to the International Space Station, an integrated defense systems, national power supply networks, transportation systems, larger infrastructure constructions.

In the first of the following three sections, the terms system and environment are discussed based on the notion of patterns, as presented in chapter 0. In the two latter chapters, the terms production system and System of Systems are discussed and defined to establish a basis for this thesis.

### 2.1.1 Definition of System and Environment

A specific system can be defined as a limited set of all possible patterns, i.e. the differences and similarities of a conceptualization, and the rest of all patterns belong to the system’s environment. This means that the boundary between the system and its environment is found where one pattern ends and another begins. This does however not mean that it is trivial to determine if a specific pattern belongs to the system or to its environment.
In order for an observer to have a complete comprehension of a system, each and every pattern belonging to the system must be conceptualized all at once. Otherwise, the observer only has a partial understanding of the system, and can only make decisions based on the patterns that are currently conceptualized. If all patterns at all scales were required to have a full understanding of the system, virtually no system could be fully conceptualized by one observer. However, an observer only needs to consider the patterns of the scales that are in effect for the observer’s specific purpose, e.g. subatomic properties are seldom in play for observers of production systems. Nevertheless, since an observer is only able to conceptualize one scale at a time, the whole system can not be conceptualized unless all necessary patterns can fit within one scale.

Each and every pattern that is possible to conceptualize

Patterns that belong to the environment

Patterns that belong to the system

Figure 4: Illustration of the definition of a system and its environment, understood through the idea that each and every possible pattern either belongs to the environment or the system.
Figure 4 is an illustration of how patterns (small hexagons) are used to define the content of a system and the boundary to its environment. The different colors of the patterns illustrate different scales, which translates to the least number of people required to fully conceptualize this system. The next driver of required individuals is the number of patterns each person is able to conceptualize at once. Consequently, in order for an observer to fully conceptualize a system, all the relevant patterns must reside within one scale and their collective information content must be lower than the capacity of the observer.

The information content can be paralleled to the number of bits required to describe the system and is consequently closely related to the complexity of the system. The information content of a system is related to two issues: system variables and system scales. The number of relevant system scales, is understood as the relevant level of detail and abstraction for a specific system or system part. For example, in a production system the molecular scale could be of environmental interest and of interest at some material processing activities; however, the molecular scale is unlikely of interest for any other point of view. Each scale is determined by a set of variables that each has a span of possible states. The total number of system states is therefore growing exponentially in a non-reducible system.

Consequently, the total amount of relevant information that one individual must handle in non-reducible system is also growing exponentially with the number of system variables.

In addition, it is important to recognize that “the sum of the complexity over all scales is the same for any system with the same number of underlying degrees of freedom (variables), even though the complexity at specific scales differs due to the organization/interdependence of these degrees of freedom” (Yaneer Bar-Yam 2004).

Consequently, it is possible to affect the degree of complexity that each person has to handle without changing the complexity of the system. The main driver of the amount of information content one person is required to handle is the architecture and organization of the system.
2.1.2 Definition of Production System

The terms *production* and *manufacturing* are commonly used as synonyms for the “processes that transform resources into useful goods and services” (Encyclopedia Britannica 2008). In more specific circumstances, one of the terms is usually reserved for “the pure act or process of actually physically making a product from its material constituents,” and the other for “all functions and activities directly contributing to the making of goods” (CIRP 2004). Unfortunately, different countries and organization have not been able to agree upon which term to use for which definition; consequently both terms can be found for both definitions.

On one hand, the International Academy of Production Engineering (CIRP) uses the term *production* as a subset of *manufacturing*, although acknowledging that *manufacturing system* is commonly used as a subset of a *production system* (CIRP 2004). On the other, a *production system* is in *Encyclopedia Britannica* defined as “any of the methods used in industry to create goods and services from various resources” (Encyclopedia Britannica 2008). Moreover, many acronyms and buzz words commonly use *manufacturing* as a subset of *production*, e.g., Toyota Production system (TPS), Flexible Manufacturing System (FMS), Mass Customization.

The etymology of *production* and *manufacturing* is more in line with the latter taxonomical alignment. The Latin roots of *produce* can be traced to *producere*, meaning “to bring forward,” and *manufacture* to *manu factus*, literally meaning “made by hand” (Merriam-Webster 1998). For the purpose of this chapter, the definition of interest is the more inclusive one, which generally includes or affects most processes within a manufacturing company or network involved in transforming resources into useful goods and services. The term used for this definition will be *production*.

The transformed resources in a production system include labor, capital (including machines and materials, etc.), and space; these resources are also labeled “men, machines, methods, materials, and money” (Encyclopedia Britannica 2008). Consequently, design of a production system includes design of “men, machines,
methods, materials, and money.” The methods used for production systems engineering should consequently address design of these resources.

Even though the term production system has been defined above, its range is too inclusive for the purpose of this thesis. Therefore, more limited definitions for specific production types must be extracted to enable a comparison with SoS. Production systems are therefore divided into three types: manufacturing system, production system, and extended production system (Figure 5). While the former is a technical system, the two latter are socio-technical systems.

A manufacturing system is here defined as a system that includes the product, the processes and the resources. In this thesis, manufacturing system concepts are divided into two classes: centralized and distributed manufacturing systems, Table 1.
Table 1: Definition and objective of five manufacturing system types, divided into the categories *centralized* and *distributed* systems.

<table>
<thead>
<tr>
<th>Type of System</th>
<th>Definition, objective, and drawbacks.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Centralized</strong></td>
<td></td>
</tr>
<tr>
<td>Dedicated Manufacturing System (DMS)</td>
<td>“A dedicated manufacturing production system is an automated system designed for the production of one product only, and which cannot readily be adapted for the production of other products” (Zhang and Alting 1994). The objective is to achieve cost-effectiveness through pre-planning and optimization (ElMaraghy 2005). A DMS is naturally unflexible, both in the long and short term.</td>
</tr>
<tr>
<td>Flexible Manufacturing System (FMS)</td>
<td>“A flexible manufacturing system is an automated system which is capable of producing any of a range or family of products, with a minimum amount of manual intervention. The flexibility is usually restricted to the family of products for which the system was designed” (Zhang and Alting 1994). The drawbacks include that the flexibility is achieved by adding functionality, which inevitably creates a suboptimal system for all specific individual tasks (Abele, Liebeck, and Wörn 2006), and that a FMS is developed with all possible functionality built in, which results in low utilization of specific functions and capital waste (Koren et al. 1999).</td>
</tr>
<tr>
<td>Reconfigurable Manufacturing System (RMS)</td>
<td>RMS “is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family”; the key characteristics of a RMS are modularity, integrability, customization, convertibility, and diagnosability (Koren et al. 1999). This solves the drawback of the general flexibility that FMS provide; however, a RMS is still designed around only one product family.</td>
</tr>
<tr>
<td><strong>Distributed</strong></td>
<td></td>
</tr>
<tr>
<td>Holonic Manufacturing System (HMS)</td>
<td>“A holonic manufacturing system can be considered as a hierarchy of self-regulating holons which function (i) as autonomous wholes in supra-ordination to their parts, (ii) as independent parts in sub-ordination to controls on higher levels, (iii) in coordination with their local environment” (Van Brussel 1994). The goal is to attain</td>
</tr>
</tbody>
</table>
“stability in the face of disturbances, adaptability and flexibility in the face of change, and efficient use of available resources” (Van Brussel et al. 1998). The granularity of HMS has in implementations become too coarse to be considered fully distributed, and implementations have become more hierarchical (Barata et al. 2007).

Evolvable Production System (EPS) EPS is based on the idea of using several re-configurable, process-oriented, intelligent modules of low granularity. This allows for a continuous adaption and evolution of the production system and the ability to explore emergent behavior of the system, which are imperative to remain fit with regards to the system environment (EUPASS 2004). The methodological approach shows in detail how an EPS is able to evolve on a module level (Figure 6).

The goal is to attain sustainability through re-engineer rather than re-development, to minimize time, and economic and ecological impact.

Figure 6: The methodology scheme for an Evolvable Assembly System (EAS), illustrates how an evolvable system is able to adapt to change on a module level (Maffei et al. 2009).

Based on their structural differences, the five manufacturing system types are classified into two main classes: hierarchical,
which includes DMS, FMS, and RMS; and distributed, which include HMS and EPS.

Even though it is mainly a technical system, a manufacturing system is here evaluated both in itself and based on its ability to function within the framework of a production system.

A production system consists of one or many manufacturing systems, as well as supporting systems, and can simplistically be understood as one production plant. Consequently, a production system usually involves all aspects relevant to product development and most of the life cycle of the production system; however, suppliers and supporting companies are considered part of the environment, not part of the system itself.

An extended production system is essentially an extended network of manufacturing companies, either with one central actor, or a network of equal actors. In these systems all aspects of product realization are of interest, including strategy, business, supply chain, and so forth. Moreover, an extended production system involves all products and product families within the network c.f. production related instantiations of SoS presented by Magee and deWeck (Magee and de Weck 2004).

2.1.3 Definition of System of Systems

In order to understand what a SoS is, it is imperative to understand what a system is. A system is a very abstract concept which in its most inclusive definition only exhibits one characteristic, the boundary to its environment:

_A delineated part of the universe distinguished by an imaginary boundary_ (Yaneer Bar-Yam 2002).

Some additional characteristics can be amended to this definition without excluding any artificial systems: a set of interacting elements, and a behavior or purpose that can not be attained by the individual elements:

_A collection of things or elements which, working together, produce a result not achievable by the things alone_ (Maier and Rechtin 2002).

It is however important to understand that the behavior, purpose, boundaries, and even existence of a system may not be
understood or agreed upon by different observers. Unless the system is simplistic, different observers’ conceptualization of a system will not overlap completely. Since the scope here is artificial systems, the lesser abstract second definition will be used in this thesis.

Even though the term SoS is tautologous and therefore taxonomically unfortunate, it is still coherent with the common usage of system within the engineering community. Within engineering, a system commonly denotes a larger more complex system, e.g. manufacturing system, airplane system, military system; and a SoS is consequently a system to which a manufacturing system, for example, is a subsystem. It is from this perspective that the term SoS should be understood.

Being a fairly new research discipline without a fully established or logical taxonomy, several SoS related terms are used as near synonyms with slightly different definitions, e.g. complex system, engineering system, socio-technical system, and enterprise system. In addition, each term is commonly only vaguely defined. In this thesis, differences or similarities between these different kinds of systems are disregarded, and SoS is, at least initially, used as a generic term for large and complex system.

Another approach to a taxonomical definition of SoS is to establish instantiations of SoS. Magee and de Weck present an extensive list of a vast range of large systems, which are classified according to societal complexity, technical complexity, and if they are natural or engineered systems (Magee and de Weck 2004). To reduce the SoS scope, neither natural systems, i.e. systems that are not man-made, nor systems that do not exhibit both a societal and technical complexity are considered SoS (labeled Engineering Systems by the authors). Systems that are related to the area of production and exhibit enough societal and technical complexity include: Automotive Products and Plants of Toyota Motor Company, Boeing-777 Aircraft System, General Motors Supply Chain, Pratt & Whitney Gas Turbine Family System. To contrast, it has to be noted that individual systems within the above mentioned do not qualify as SoS, e.g. Boeing-777 in itself and a specific line of Toyota cars.
A number of attributes were used to assess the above mentioned list of SoS. From this assessment, the following characteristics were found to be typical of all SoS, i.e. can not be used to differentiate amongst SoS: all SoS are:

Complex, real, open, artificial, dynamic, hybrid (system states are both continuous and discrete) and have mixed control (have both autonomous and human-in-the-loop elements or subsystems. (Magee and de Weck 2004).

This definition should be contrasted with the following set of SoS, SoSE and Complex System definitions:

1. Large scale concurrent and distributed systems that are comprised of complex systems (Kotov 1997).
2. SoSE involves the integration of systems into systems of systems that ultimately contribute to evolution of the social infrastructure (Lukasik 1998).
3. Systems of systems are large-scale concurrent and distributed systems that are comprised of complex systems (Carlock and Fenton 2001), (Jamshidi 2005).
4. Systems of systems are operationally and managerially independent, exhibit evolutionary development and emergent behavior, and are geographic distributed (Maier 1996), (Sage and Cuppan 2001).
5. The two most important characteristics of complex systems are: self-organization and that they arise through evolutionary processes (Minai, Braha, and Y. Bar-Yam 2006).
6. SoS exhibit five distinguishing characteristics: autonomy, belonging, connectivity, diversity, and emergence (Boardman and Sauser 2006).
7. SoS are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal (Jamshidi 2008).
8. According to Norman and Kuras (Norman and M.L. Kuras 2004), a complex system is a system:
   a. Whose structure and behavior is not deducible, nor may it be inferred, from the structure and behavior of its component parts;
b. Whose elements can change in response to imposed “pressures” from neighboring elements;

c. Which has a large number of useful potential arrangements of its elements;

d. That continually increases its own complexity given a steady influx of energy;

e. Characterized by the presence of independent change agents.

9. SoS exhibit the following characteristics: evolutionary behavior, self-organization, heterogeneity, emergent behavior, and that SoS are small-world and scale-free networks (M. Bjelkemyr, Semere, and B. Lindberg 2009).

The generic properties highlighted in each definition are naturally dependent on the set of SoS studied. Given the vast number of different SoS instantiations, every researcher will naturally find properties that are similar to those found by others, but not the same. However, the characteristics used in the definitions above are synonymous or at least topically related: self-organization (operationally and managerially independent, mixed control, autonomy, and belonging); evolutionary behavior (concurrent is not synonymous but often a prerequisite for evolution); heterogeneity (diversity, distributed); small-world and scale-free networks (connectivity).

To corroborate the hypothesis, it is practical to use a definition of SoS where individually verifiable SoS characteristics are amended to a generic definition. A definition that includes the most important and predominant characteristics of SoS is used in this thesis:

A large and complex artificial system that exhibit the following characteristics: self-organization, evolutionary behavior, heterogeneity, emergent behavior, and that SoS are complex networks.

To distinguish systems that exhibit these characteristics from those that do not, the term SoS subsystem is introduced. The term is in this thesis not used for classification, it is merely a term for systems that appear similar to SoS, but only exhibit few or none of the complex SoS properties (Figure 7).
2.1.4 Characteristics of System of Systems

Self-Organization

Self-organization is here defined as “a process where order emerges without external control based on local interactions of constituent components.” (NECSI Wiki 2008a). Self-organization is similar to evolution, but where evolution primarily takes place in a system’s interface to its environment; self-organization is an internal system process that is not “being controlled by the environment or an encompassing or otherwise external system” (Heylighen 1997a).

Self-organization is a common property of natural systems, where forces commonly act over short distances; on the contrary, it is rarely a spontaneous property of artificial systems. However, both the system design process and the continuous improvement process in Kaizen are to some extent self-organizational (Minai, Braha, and Y. Bar-Yam 2006).

For socio-technical systems, self-organization can be decomposed into operational and managerial independence. Operational independence signifies that SoS subsystems are independent and useful in their own right. Managerial independence signifies that a system is both able to operate independently and actually is operating independently (Maier 1998).
Evolutionary Behavior

Evolution is commonly understood from a biological context as a “trial-and-error process of variation and natural selection of systems” (Heylighen 1997b). However, the Darwinian trial-and-error process is in engineering commonly guided by a goal of specific intent. The definition used in this thesis is therefore:

*a guided “process of variation and natural selection of systems where selection is automatic and a result of the internal or external environment”* (Heylighen 1997b).

The life cycle of SoS subsystems are usually not evolutionary; partly because they are regularly encapsulated within the traditional life-cycle stages; and partly because the decision structure is commonly hierarchical. Since SoS subsystems are not synchronized, the SoS itself is continuously and iteratively going through all stages at the same time. During evolution of a SoS, unlike Darwinian evolution, “a [system] configuration can be selected or eliminated independently of the presence of other configurations” as long as the configurations are not subsequent system states (Heylighen 1997b).

Heterogeneity

SoS consist of a multitude of dissimilar or diverse subsystems, structures, relationships and agents, i.e. nodes, links and network properties in the system are of multiple types. Heterogeneity is naturally a strong driver of system complexity since the description of two different types of nodes requires a more extensive description than two identical nodes; cf. Kolmogorov complexity – the absolute minimal length of a computer program required to describe a system.

A system is often heterogeneous on multiple layers simultaneously, e.g. size, architecture, life-cycle, scientific area, and elementary dynamics. This increases the difficulty of modeling a SoS and requires people from different knowledge and science domains to work side by side. As a result, new demands for communication and information handling are required, i.e. rules for interactions between the interfaces of nodes and systems in a system.
Emergent Behavior

Emergence is the added behaviors that arise due to the interactions between a system’s parts and sub-systems, and that cannot be directly attributed to any individual sub-system, part or trivial relationship between these. Emergent phenomena are therefore in this thesis understood as the patterns that result from interactions among elements in a system and their interactions with the environment (A. Clark 2000). In other words, emergence “refers to how behavior at a larger scale of the system arises from the detailed structure, behavior and relationships on a finer scale (NECSI Wiki 2008b)”. The definition used in this thesis does not belong to any of the two varieties of emergence that are commonly discussed: weak emergence and strong emergence, it is rather in between them and can be labeled moderate emergence. Weak emergence denotes a macroscopic system state that can be derived from microscopic system states, but only through extensive modeling and simulation. Strong emergence denotes that high-level behaviors are autonomous from the systems and elements on lower levels (Bedau 1997). These two kinds are often intertwined, which creates confusion, especially regarding how emergence can be dealt with (Johnson 2006). Reduction of weak emergence is a substantial part of engineering of reducible systems, and the engineer must always prioritize between knowledge of system behavior on the one hand, and time and resources spent on modeling and simulation on the other.

The definitions used in this thesis, and in most engineering and complex systems disciplines, it is the weak emergence that is of primary interest.

Complex Networks (Small-World and Scale-Free)

Social systems are commonly analyzed from a network perspective, where individuals are reduced to nodes that are either connected or not depending of there is a particular relationship between them. This simplification of a social system enables an analysis of the structure of the network, free of superfluous information about each individual or the type of relationship two
individuals have. A network analysis can both provide predictions of how a system will function in a particular scenario, and enable alterations to the network structure to avoid or obtain a certain system feature.

A network is usually represented by a graph with a set of nodes (or vertices or points) that are connected by a set of edges (lines or arcs). Both nodes and edges can be of one or many types (modes), and the connecting edges can be either directional or non-directional. A system is naturally a combination of multiple kinds of types of nodes and edges, and the connections include both non-directional and directional lines. Depending on the topology of the nodes and edges in a network, different kinds of network properties emerge. The approach of focusing the attention on the relationships between nodes, instead of the properties of individual nodes, enables observers to better understand the dynamics of a system.

There are two models of networks that are of particular interest when studying SoS: small-world networks and scale-free networks.

In a small-world network few nodes are directly connected to each other, but most nodes can be reached from every other in a small number of steps. This means that a small-world network is positioned in between a completely regular and a completely random network (Figure 8).

![Figure 8: Small-world in comparison to regular and random models of networks (Watts and Strogatz 1998).](image-url)
Unlike a regular network, where all nodes are equally connected to its neighboring nodes, some of the regular edges are random instead. The path between two random nodes is therefore short in comparison to a similar regular network. This results in that a dynamic system with small-world coupling displays “enhanced signal-propagation speed, computational power, and synchronizability” (Watts and Strogatz 1998).

In a scale-free network the number of edges of all nodes in the network are inhomogeneous and follow a power-law distribution, i.e. while most nodes have few edges; some nodes are highly connected and function as hubs in the network. This should be contrasted to an exponential network where most nodes are equally connected. A common example is the relationship between the magnitude and the occurrence of earthquakes: there are few high magnitude earthquakes; considerable more medium sized earthquakes, and a lot of small earthquakes. A consequence of a network exhibiting a power-law distribution is that they are fault tolerant to random failure, but at the same time they are vulnerable to a focused attack on the hubs (Figure 9).

Figure 9: Comparison between an exponential and a scale-free network (Albert, Jeong, and Barabasi 2000).

Both small-world and scale-free networks are the result of three statistical network characteristics: average path length, clustering coefficient, and nodal degree. Average path length ($L$) is defined as the average least number of steps between any two nodes in a network; and it is a measure of the efficiency of a network. With a
shorter average path length, fewer steps are on average required to
distribute information or mass inside the network. The clustering
coefficient \( C \) is defined as the probability that two nodes, which
both are connected to a third node, also are connected to each
other. Consequently, the clustering coefficient is a measure of the
network’s potential modularity (Braha and Y. Bar-Yam 2004a).
The third characteristic, nodal degree, is the distribution of the
number of edges for all the nodes in a network.

To determine if a network is a small-world the statistical
characteristics are related to those of a random network with the
same number of nodes and edges. Two criteria should be fulfilled:
the network’s average path length should be similar to that of a
random network \( (L_{\text{actual}} \approx L_{\text{random}}) \), and the network’s clustering
coefficient should be greater than that of a random network
\( (C_{\text{actual}} > C_{\text{random}}) \). For example, vehicle design has been shown to
be a small-world network (Table 2).

Table 2: Example of network measures (excerpt from Table 6)

<table>
<thead>
<tr>
<th>Network</th>
<th>N (number of nodes)</th>
<th>K (average degree)</th>
<th>( L_{\text{actual}} )</th>
<th>( L_{\text{random}} )</th>
<th>( C_{\text{actual}} )</th>
<th>( C_{\text{random}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Design (*)</td>
<td>120</td>
<td>6.95</td>
<td>2,878</td>
<td>2,698</td>
<td>0.205</td>
<td>0.07</td>
</tr>
</tbody>
</table>

To be considered as a scale-free network the degree distribution
should follow a power law distribution, i.e. while a small number
of nodes are highly connected hubs; most nodes are considerably
less connected Figure 10.

![Figure 10: A power law distribution; y-axis denotes the nodal degree, x-axis denotes the number of nodes with a particular nodal degree.](image)
2.2 Systems engineering

Systems engineering (SE) is here defined as the direct or indirect creation and evolution of an artificial system. In this thesis, systems engineering is decomposed into two subcategories: reducible systems engineering (RSE) and system of systems engineering (SoSE). These two are considered partially overlapping, but are used for different types of systems (B.E. White 2006). RSE is what most people understand as systems engineering, and is unlike SoSE appropriate for systems that can be reduced to the sum of its components, i.e. systems that do not exhibit emergent behavior. However, it is also a common approach for non-reducible systems because of the difficulty in understanding emergence and that it has an effect on engineering. SoSE is developed for evolving and emergent systems, but would generate suboptimal reducible system solutions, both with regards to quality and project time and cost.

2.2.1 Reducible Systems Engineering

Reducible systems engineering is a process of transforming stakeholder requirements into a system that has the ability to answer to stakeholder needs and wants. A successful RSE process is characterized by high system quality and performance, low development costs, and short time-to-market (K. B. Clark 1989). Improvements of the design process should consequently focus on these issues. Failure to achieve an efficient engineering process leads to “long lead time, […] poor quality of systems design, and an ‘over the wall’ mentality – resulting in mis-investment and inefficient manufacturing systems” (Fritz et al. 1994).

Design of a system can often be decomposed into three main tasks: decide what to do, decide how to do it, and finally doing it. The two initial tasks, what and how, are the basic questions that systems engineering methods aim to help answering. Neither of these questions is trivial, and the quality of the final system is dependent on all three tasks.

In RSE, what is answered by a set of interrelated system requirements, which usually can be derived from stakeholder needs. From a RSE perspective, the ideal is to develop a closed set
of complete, detailed and precise requirements (Norman and M.L. Kuras 2004). In general, this is achieved through five core requirements engineering activities: eliciting requirements, modelling and analysing requirements, communicating requirements, agreeing requirements, and evolving requirements (Nuseibeh and Easterbrook 2000). These activities naturally become more complex with an increase of interrelated support systems that pose both direct and indirect requirements.

In reality, the sequence of establishing requirements and then fulfilling the requirements is not as strict as in the description above. In order to develop lower level functional requirements, a solution to higher level requirements must already be established. This leads to a design process that iteratively addresses system requirements and system solutions, cf. the spiral development of functional requirements (FRs) – design parameters (DPs) – and process variables (PVs) (Suh 2001). For a system that is fully developed within one organization, this is not necessarily a problem; however, the iterative design process poses a problem for the development of requirement specifications and for the evaluation of suppliers’ system solutions (Jeziorek et al. 2005).

According to the ISO/IEC 15288 standard, the whole RSE design process is made up of sequential and parallel activities that commonly can be placed within any of 25 well defined process types defined in the standard (ISO 2002) (Figure 11). To exemplify, machining and assembly are both “Technical Processes”, and are sub-processes of the “Operation Process”. The processes within the standard are not linked to each other; they rather function as modular blocks that are individually configured for each RSE project. This configuration is a key task in the initial stages of systems engineering. For simpler systems, predetermined generic design processes are commonly used. This reduces project time and cost, but the key reason is to assure a certain level of quality to the systems engineering process, and thereby to the system that is engineered too.
Figure 11: System life cycle processes of ISO/IEC 15288, divided into four key areas: enterprise, agreement, project, and technical processes (ISO 2002)

### 2.2.2 Systems Engineering Methods

Traditionally, systems engineering has been an experience based task, making it vulnerable to personnel turnover, inflexible with regards to system size and project time, and difficult to improve efficiency of the process. With increased competition, the risks associated with a mainly experience based design process are too high, and failure may lead to severe economic loss and market share decline. In addition to loss of knowledge and information through personnel turnover and an insufficient knowledge management system, an experience based design process provides insufficient means to fully comprehend most system to a required degree. This is related both to the information content of the system and to the multiple scales being required to fully conceptualize the designed system (M. Bjelkemyr and B. Lindberg 2007). It is required to simultaneously cope with a diverse set of technical, natural and organizational issues; failure to do so will result in emergent effects caused by system features that were outside of the designers’ conceptualization of the designed system.
Standardized tools and methods are used during production systems engineering to improve efficiency and the end result. Standardization of methods provides a common mean of communication, reduces planning, assures that all key aspects are considered, and enables improvements to be made on the methods themselves. However, inappropriately used and implemented methods often have the opposite effect, e.g. an initial increase in workload, may create a false sense of security, and in an attempt to cover all aspects of a system the tools tend to swell and become difficult and awkward to use (Fritz et al. 1994).

On one hand, there is a need for tools and methods; on the other, there is a problem in achieving an appropriate scope for a generic method and to make it user-friendly. These conflicting positions have been substantiated in both case studies where master students were to concurrently develop a model of a product and its production system using KTH-IPM (Figure 13), (D. Aganovic and M. Bjelkemyr 2004); and in the development of a model driven framework for production systems engineering in collaboration with Scania AB and several other Swedish production companies (Figure 12), (ModArt 2009).

Figure 12: Illustration of the high-level processes in the ModArt aid for model driven engineering of a manufacturing system (ModArt 2009).
For design of non-basal technical systems, multiple activities are required to solve the necessary task of transforming requirements to a design solution. These activities are, based on their input and output, logically arranged to both minimize interdependence between activities and reduce non-value adding activities. This arrangement of activities can then be used as a schedule for what to do and when, and also amended with methods and tools to create a framework to be used during design of a system. Examples of systems engineering frameworks include: ModArt (ModArt 2009), KTH-IPM (D. Aganovic 2004), Manufacturing System Design Framework Manual (Vaughn, Fernandes, and Shields 2002), and Design for Six Sigma (Yang and EI-Haik 2003).

These frameworks are mainly focused on the development of one single system, although some of them address concurrent engineering of a product and its manufacturing system. This is partly a consequence of keeping the framework manageable and user-friendly, and partly to keep it generic and applicable to a wide variety of systems. Also, by keeping the structure of the framework similar to that of reducible systems engineering, the introduction of the framework becomes fairly seamless. Unfortunately, most engineering systems are closely related to at
least one other system, which is difficult to address within the frameworks, cf. discussion on process variables (PVs) and concurrent engineering in Axiomatic Design (Suh 2001) and (D. Aganovic 2004). To successfully engineer and re-engineer multiple system, a model driven engineering environment that is based on reference architecture is therefore required.

2.2.3 RS and SoS Differences, and Effect on SoSE

A key reason for distinguishing SoS from less complex systems is because RSE becomes inappropriate at a certain level of system complexity. The reasons for this are multilayered and can be attributed to fundamental system concepts regarding both a system and engineering of that system, Table 3.

Table 3: Comparison between SoS and RS, adapted from (M. Bjelkemyr, Semere, and B. Lindberg 2009) and (Norman and M.L. Kuras 2004), and the effect on SoSE.

<table>
<thead>
<tr>
<th>Reducible Systems (RS)</th>
<th>SoS</th>
<th>Consequence for SoSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>- RS are reproducible.</td>
<td>- No two SoS are alike.</td>
<td>- Each SoSE project must be individually designed; best practice cannot be applied.</td>
</tr>
<tr>
<td>- RS are realized to meet pre-conceived specifications, i.e. reduce complexity.</td>
<td>- SoS continually evolve to increase their own complexity.</td>
<td>- SoSE should develop a framework that is able to continuously evolve a system.</td>
</tr>
<tr>
<td>- RS have well defined boundaries.</td>
<td>- SoS have ambiguous boundaries.</td>
<td>- The system’s environment becomes the main subject.</td>
</tr>
<tr>
<td>- Unwanted possibilities are removed during the realization of RS.</td>
<td>- New possibilities are constantly assessed for utility and feasibility in the evolution of SoS.</td>
<td>- SoSE is open-ended - Requires a different engineering mindset</td>
</tr>
<tr>
<td>- External agents integrate RS.</td>
<td>- SoS are self-integrating and re-integrating.</td>
<td>- SoSE creates a development framework</td>
</tr>
<tr>
<td>- Development always ends for each instance of RS realization.</td>
<td>- SoS development never ends, SoS evolve.</td>
<td>- In SoSE, development and operations are intertwined.</td>
</tr>
<tr>
<td>- RS development ends when unwanted possibilities and internal friction are removed.</td>
<td>- SoS depends on both internal cooperation and competition to stimulate evolution.</td>
<td>- SoSE must provide an internal market mechanism, which enables agents to interact within a framework</td>
</tr>
</tbody>
</table>
The main difference between RSE and SoSE is not the engineering process in itself, but what is being engineered. The purpose of RSE is to engineer the end-product, and the purpose of SoSE is to engineer a framework that is able to evolve the intended product in accordance with environmental changes. In addition, SoSE is an open ended task where the development and operation of the system are parallel tasks. To enable distributed control, each agent must be able to negotiate by itself.

2.2.4 Decision Making in SoSE

To further illustrate the key distinctions between different engineering contexts, RSE and SoSE are positioned within the Cynefin framework, which addresses decision making in different environmental contexts (Figure 14), (Snowden and Boone 2007).

![Figure 14: The Cynefin Framework describes that the appropriate response to a situation is related to the engineering context, i.e. simple, complicated, complex or chaotic (Snowden and Boone 2007).](image)

The purpose of the framework is to help decision makers determine the type of context they are in, and the appropriate activities to perform in order to make correct decisions. In the framework, there are five different decision environments: simple, complicated, complex, chaotic, and disorder; where the last denotes that it is unclear which of the other four environments is predominant.
Each decision context calls for different actions from the leaders, and must therefore be identified prior to any actual decisions can be made. This is also the case for decision making in SE, where an inappropriate approach will result in anything from a suboptimal system solution to project catastrophe and severe societal effects; cf. (Y. Bar-Yam 2003a).

In Table 4, the four key decision environments are characterized and related to a decision action plan. When compared to the above descriptions of RSE and SoSE in Table 3, it is evident that they belong to different decision making contexts.

Table 4: Characteristics of the four engineering contexts and main tasks of the leader, extracted from (Snowden and Boone 2007)

<table>
<thead>
<tr>
<th>Context Characteristics</th>
<th>Task of Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple</strong></td>
<td>- Sense, categorize, respond</td>
</tr>
<tr>
<td>- Repeating patterns and consistent events</td>
<td>- Ensure that proper processes are in place</td>
</tr>
<tr>
<td>- Clear cause-and-effect</td>
<td>- Use best practice</td>
</tr>
<tr>
<td>- One correct answer</td>
<td></td>
</tr>
<tr>
<td>- Fact-based management</td>
<td></td>
</tr>
<tr>
<td><strong>Complicated</strong></td>
<td>- Sense, analyze, respond</td>
</tr>
<tr>
<td>- Expert diagnostics required</td>
<td>- Create panels of experts</td>
</tr>
<tr>
<td>- Discoverable cause-and-effect relationship</td>
<td>- Listen to conflicting advice</td>
</tr>
<tr>
<td>- Multiple correct answers</td>
<td></td>
</tr>
<tr>
<td>- Fact-based management</td>
<td></td>
</tr>
<tr>
<td><strong>Complex</strong></td>
<td>- Probe, sense, respond</td>
</tr>
<tr>
<td>- Flux and unpredictability</td>
<td>- Enable patterns to emerge</td>
</tr>
<tr>
<td>- No right answer, emergent instructive patterns</td>
<td>- Increase levels of interaction and communication</td>
</tr>
<tr>
<td>- Many competing ideas</td>
<td>- Use idea generating methods</td>
</tr>
<tr>
<td>- Pattern-based leadership</td>
<td></td>
</tr>
<tr>
<td><strong>Chaotic</strong></td>
<td>- Act, sense, respond</td>
</tr>
<tr>
<td>- High turbulence</td>
<td>- Look for what works instead of right answers</td>
</tr>
<tr>
<td>- No clear cause-and-effect, no point in looking</td>
<td>- Take immediate actions to reestablish order</td>
</tr>
</tbody>
</table>

RSE and operation of reducible systems exhibit characteristics from both the simple and complicated contexts, though the engineering of system issues clearly belong to the complicated domain. Consequently, both decision contexts are based on facts.
that can be found out. RSE is a process that generates single or multiple solutions, and has a clear or at least discoverable link between cause and effect. The decision process should, according to the Cynefin framework, therefore be: Sense – Categorize/Analyze – Respond, which is inline with the current practice of RSE.

SoSE is closely related and pose similar decision contexts as the operation of a complex system, i.e. SoS. Decisions in SoSE can not be based on facts in the same sense as for RSE. The reason for this is further discussed in chapter 3.2, but it is based on the impossibility to obtain and process the relationship between all necessary facts. Decisions in SoSE must therefore be based on patterns, i.e. similarities and differences that can be recognized on a higher level of abstraction. Both complex and chaotic decision contexts are pattern-based and are consequently of relevance for SoSE; however, in a chaotic context the purpose is not to evolve the system, rather it is to make sure that the system survives and can go back to its complex state. Consequently, the purpose of SoSE is to create a framework that facilitates detection of patterns, increases interaction between all parts of the system, and that is based on a Probe – Sense – Respond decision process.

2.2.5 A Biological Approach to SoSE

Due to their evolutionary characteristics, SoS are often paralleled with natural systems, particularly with regards to transferring the development and evolution of biological systems on to engineering of SoS. Every biological organism has a genome that contains all the necessary biological information for building and maintaining an organism. However, the genome has principally a passive role in the development of an organism; the actual development is carried out by proteins. Consequently, an organism has both a genome complexity and a complexity that is related to the environmental effects during the development into an actual organism.

Similarly, engineering is based on a reference architecture, which like the genome contains all the necessary information for building and maintaining that system, i.e. for interacting with the system’s environment. Depending on the specific state of the system and its environment, specific parts of this information is
<table>
<thead>
<tr>
<th>Biological development (Banzhaf and Pillay 2007)</th>
<th>Relevance to the SoSE framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>“The genome is merely used to channel or canalize environmental complexity”.</td>
<td>The SoSE framework should be used to channel or canalize the complexity of its environment.</td>
</tr>
<tr>
<td>“Fitness of the organism must […] be incremental, i.e. a primitive function is required from the very beginning”.</td>
<td>The SoSE framework must both enforce and handle incremental development.</td>
</tr>
<tr>
<td>“The development process is robust and fault tolerant and has to include repair mechanisms, with the possibility of regeneration, and with adaptability”.</td>
<td>The SoSE framework itself must be fully robust and fault tolerant to changes in its environment. Failure to any part of the SoS itself should not generate catastrophic system failure.</td>
</tr>
<tr>
<td>“If the required complexity cannot be put in place in one go, it needs to be grown”.</td>
<td>A RS can be engineered in one go; a SoS must be grown, orchestrated by a SoSE framework.</td>
</tr>
<tr>
<td>“mapping a genome […] into a dynamic process expressing itself as growth and behavior is as fundamental to an organism as [the genome itself]”.</td>
<td>SoSE consists of two frameworks: one that facilitates the development of a specific system genome; and one that facilitates the process of transferring the system genome information into an actual system.</td>
</tr>
<tr>
<td>“The time dimension has further implications as a way to ‘mold’ results of development”</td>
<td>The result of SoSE is not only dependent on what is done and how, but also when.</td>
</tr>
<tr>
<td>“Growth is closely related to duplication and divergence”.</td>
<td>Both the SoSE and the CS itself should be growing incrementally.</td>
</tr>
<tr>
<td>“While specification is irreversible […] there is no isolation of parts”.</td>
<td>CS subsystems should not be developed in isolation; their interfaces should be continuously negotiated and their fitness regularly evaluated.</td>
</tr>
<tr>
<td>“Development allows the exploitation of side-effects”.</td>
<td>SoSE must facilitate the development of unanticipated advantageous possibilities.</td>
</tr>
<tr>
<td>“Open-endedness of the process of evolution is allowed”.</td>
<td>Any and every part of the system environment evolves; therefore, the system must also be able to evolve and adapt in any direction.</td>
</tr>
</tbody>
</table>

used to further adapt and evolve the system in accordance with the environment. The purpose of SoSE is consequently twofold: to provide a framework for the development of the reference architecture; and to provide an engineering framework for the process of transferring the reference architecture into an actual system. This is the basis for how biological development is related
to SoSE in this thesis. In Table 5: Traits of biological development and its relevance to SoSE. Table 5, some additional key aspects of biological development (Banzhaf and Pillay 2007) and their relevance for SoSE are presented to further advance the understanding of this relationship.

2.3 Conceptualization Framework

The framework is in the thesis used to understand the limitations of an individual, and thereby the absolute limitations of the systems engineering process. The framework is loosely based on concepts presented by Kuras (M. L. Kuras 2006). Since the architectural structure of production systems engineering has grown out of the engineering process for less complex products, it is not adapted for engineering of the more complex SoS. This issue is further developed in the test of hypothesis B.

The purpose of using the multi-scale framework is to recognize the individual’s role in both research and systems engineering. This enables a better understanding of what can and cannot be achieved within these areas, and thereby enables researchers and system engineers to circumvent their limitations to achieve their goals.

To address what a conceptualization is and how it is formed, a framework based round the trinity of the observer, reality, and conceptualization is presented in the form of a concept model where different concepts are related to each other (Figure 15). The term conceptualization needs to be clarified to avoid confusion. The conceptualization is a mental model that the observer gets when studying the reality, and is consequently a noun. Naturally, this mental model is closely related and affected by the process of conceptualizing.

The framework is based on the assumption that conceptualizations are based on the reality but distinct from it. An observer must acknowledge that other observers do not necessarily share exactly the same conceptualization of a particular part of the reality. In a person to person interaction, divergent conceptualizations are commonly aligned through incremental adjustments; something that is difficult in a computer to computer interaction. In
engineering the issue of representation has been addressed through for example ontologies and standardization of how products, processes and resources are represented The issue of unambiguous representation is a key issue in requirements management, where unambiguous interpretation is imperative in order to develop a solution that answers to the intended requirements.

Figure 15: Concept model of the substance of a conceptualization.
2.3.1 The Substance of a Conceptualization

The observer studies the reality and forms a conceptualization, which is based on but distinct from the reality. The reality has both an extent, and a richness, which can be understood as the width and depth of the reality. The extent and richness of the reality correspond to the concepts field of view, i.e. the expanse of the conceptualization; and resolution the degree to which a part of the conceptualization can be distinguished.

Due to limitations to an observer’s brain there is a finite amount of information in any given conceptualization. The resolution and field of view are consequently linked and may restrict each other. The coordinates beyond an individual’s capacity, are impossible to conceptualize. Illustrated in Figure 16 by a conceptualization of an olive tree: starting from top left, (1) the observer conceptualizes the whole tree with limited detail; (2) if the field of view is limited the observer only conceptualizes a limited part; (3) by increasing the resolution the observer can conceptualize the details of leaves, twigs, and olives; (4) if the observer desires to conceptualize a larger part of the olive tree again, the resolution must be gradually decreased.

Figure 16: Illustration of how the correlation between field of view and resolution affect our ability to conceptualize everything at once.
**Content**: the substance of the conceptualization. The content elements are the smallest piece of information available, and can in a visual sensory perception metaphorically be understood as pixels, the smallest piece of information in a picture. However, content elements are not merely two-dimensional squares; but can belong to any of the following perceptive functions: language (phonological, lexical, morphological), time (duration, temporal order, time perspective), place (spatial distance, spatial direction, spatial location), response (egocentric, feedback from motor response, response selection), reward value (positive – negative), sensory perception (height, color, shape, orientation, motion, contrast) (Kesner and Martinez 2007)

The content is composed into **patterns**, i.e. “similarities and differences that we recognize in a conceptualization” (M. L. Kuras 2006). Depending on the content, field of view, and resolution, different patterns will appear (Figure 17). The use of patterns is an effective technique to minimize the information content in a conceptualization by not having to consider every single content element in a conceptualization, and thereby increasing the available field of view/resolution coordinates. It is important to stress that patterns are merely a set of content elements that an observer has perceived. This set of elements has not been labeled; i.e. an observer has taken in the content information but not yet related it to prior knowledge or interrelated patterns.

![Figure 17](image-url)  
**Figure 17**: Illustration of how resolution and field of view may change which patterns are visible for the observer (M. Bjelkemyr and B. Lindberg 2007).
The next step in forming a conceptualization is to attach meaning to the patterns. Everything that has a meaning can potentially be labeled with a word; however, far from every meaning has been labeled, cf. the different number of words and mismatches between different languages. How this meaning is attached or what it is goes beyond the purpose of this thesis will not be further discussed.

The final property of a conceptualization is termed scale, a concept defined as the “level of detail visible to an observer of a system” (Y. Bar-Yam 2002). It is used for when the current set of patterns is changed, i.e. some patterns are lost and some are gained. An observer is only able to conceptualize one scale at a time; consequently, the observer is unable to think about the patterns that have been lost unless the observer returns to the previous scale. A very simple example of this feature is a so called bi-stable figure (Figure 18).

![Figure 18: Bi-stable figure of Rabbit/Duck (Jastrow 1899)](image)

### 2.3.2 Limitations to Conceptualizing

The ability of an individual is related to that individual’s brain and the number of neurons in the brain can be related to the number of bits of information the brain can process at the same time. This is the absolute limit of an individual’s ability to conceptualize (Y. Bar-Yam 2003b). The information content of a system increases exponentially, i.e. the number of bits required to fully understand a system doubles for each new variable that is introduced.

This provides an interesting background for understanding an individual’s limitations; the main idea is however that there is a limit to our abilities and that the limit is much lower than that of...
most production systems. For the purpose of this thesis, two key ideas are highlighted, and will be further advanced in chapter 3.2, which presents the results of hypothesis B:

The capacity of the human brain and sensory organs are finite. The observer is consequently only able to conceptualize a limited part of the reality at the time, and must make an adjustment to be able to conceptualize another part of the reality. In the engineering of SoS, this adjustment of field of view is continuous. The capability is naturally different between individuals, and an individual can improve the ability to only focus on relevant issues; however, the complexity of a SoS far exceeds the capability of an individual. This leads to that multiple individuals are required to engineer a SoS, which in turn introduces the above described problem with ambiguous conceptualizations.

An observer is only able to conceptualize at one scale at the time. This means that an observer is not able to hold two diverse perspectives at the same time, and is thereby not able to compare multiple perspectives without changing the conceptualization back and forth. For example, geometric details, high level system properties, and economical issues of a production system can not be conceptualized at the same time by one observer. The effect is that an observer is not able to fully understand even a small system, which results in that engineering decisions are based on a simplistic understanding of interrelated conceptualizations. This limitation can be partially bridged by using engineering methods that can interconnect multiple scales of a system. An illustrative example is DFX-tools, where requirements from an interrelated area, e.g. assembly or manufacturing, are transformed into simple design rules that can be conceptualized at the same scale as other design requirements.
3 Results

In the following chapter the two hypotheses are individually described and tested in detail. The section for each hypothesis ends with a discussion on whether it has been corroborated or falsified. The answer to the main research question is derived from testing the two hypotheses; this is further discussed in the final part of this chapter. However, it is important to keep the main research question in mind in order to fully appreciate the relevance of hypothesis A and B.

Does Reducible Systems engineering sufficiently address all production system characteristics, or is a new approach required?

3.1 Hypothesis A

Production Systems exhibit System of Systems Properties.

Hypothesis A states that a production system is a system of systems, i.e. a production system exhibits a majority of the properties that SoS do. The test of this hypothesis is achieved in two steps: first the terms “production system” and “system of systems” are defined individually (Chapter 2.1); second, the definitions are compared to establish common grounds and discrepancies (Chapter 3.1.1); finally the results are discussed, including a discussion of if the hypothesis has been corroborated or falsified (Chapter 3.1.2).

3.1.1 SoS Characteristics in Production Systems

For a production system to be considered a SoS, the production system should exhibit most of the SoS characteristics. Each characteristic has been evaluated, mainly through literature studies, to determine to what degree it is a property that different types of production systems exhibit. The evaluation is argumentative to its nature and the degree to which a SoS properties can be exhibited in a production system is an evaluation ranging from not occurring (-) to strong occurrence (+++). The evaluation is affirmative in the sense that occurrence of a characteristic, especially one that is not commonly
understood as a property of that system, is more commonly reported than the lack of that characteristic. However, for some of the SoS characteristics there are paradoxical properties, leading to a possible conclusion of non-occurrence of a SoS characteristic.

Each SoS characteristic is introduced with the definition used in this thesis to remind of the key elements of that property. At the end, a table presents the final evaluation of occurrence of SoS characteristics in production systems. These tables are then merged and discussed in the succeeding Chapter 3.1.2.

**Self-Organization**

“a process where order emerges without external control based on local interactions of constituent components” (NECSI Wiki 2008a).

Self-organization is an internal system process described as “the spontaneous appearance of large-scale organizations through limited interactions among simple components” (Minai, Braha, and Y. Bar-Yam 2006). It is a common property in natural systems based on that only relatively closely located nodes can be interlinked. In artificial systems, links between nodes are not necessarily decided by location, rather by a superimposed organizational structure. In engineering companies, a majority of organizational structures are hierarchical, e.g. functionally decomposed, project based, or matrix structure in which multiple hierarchical structures superposed each other. However, both the systems engineering process and the continuous improvement process in Kaizen may be considered self-organizational due to a bottom up approach that is guided by distributed rules rather that a centralized control system (Minai, Braha, and Y. Bar-Yam 2006).

For socio-technical systems a general distinction is made between operationally independence, i.e. a modular system in which sub-systems are able to operate independently; and managerial independence, i.e. a system in which sub-systems are both operationally and managerially autonomous. Even though operationally independent systems are not self-organizational, they possess the ability to adapt to self-organization if required.

Extended production systems are on a first tier level generally composed of operationally independent subsystems, i.e. modules
that are both able to function individually at a local level. First tier subcontractors and suppliers are naturally both operationally and managerially independent, and are constantly negotiating their role in both the extended production system and in other non-related systems. In Figure 19, Swedish suppliers to the automotive industry located in the vicinity of Gothenburg are shown to demonstrate the size of the local supply chain (Djurberg and Backlund 2008), and to illustrate how global effects have local effects for a region that is strongly dependant on the automotive industry.

Figure 19: Automotive suppliers in the Gothenburg area (detail from (Djurberg and Backlund 2008)).

Similar to an extended production system, the subsystems within a regular production system are commonly organizationally independent. Divisions within a single company are able to function by themselves; however, they are seldom managerially independent. This means that a division can be outsourced by the company, but is not able to decide whether to remain a part of the company. There are however examples of self-organizing production system, e.g. SEMCO a company that has gone from
manufacturing centrifuges to providing a wide variety of both products and services. The transformation was enabled by introducing small self-managed groups that are fully accountable and responsible for all aspects within a set of products (Killian, Perez, and Siehl 1998).

During operation, traditional manufacturing systems are usually controlled through two kinds of processes deciding how and when a product is going to be manufactured: manufacturing process management and enterprise resource planning. These central control systems exclude self-organization. In addition, sub-systems and components do not possess the ability to decide what they are going to manufacture and when. Even though human operators are part of the system, central control takes precedence and only limited self-organization is usually permitted.

A key prerequisite for improving the autonomy of a technical system is that the processes are stable (Figure 20). To become an autonomous system, a fully self-reconfigurable system must exhibit the first four stages to become able to handle unforeseen situations (Maffei et al. 2009). In addition, “artificial systems […] may require a kind of leader, a broaker or […] decision maker” (Barata et al. 2007), which leads to that an artificial system may be partly self-organized, e.g. operationally autonomous but centrally controlled when it comes to strategic issues.

![Figure 20: Relation between the stability of a system's processes and the obtainable autonomy (Maffei et al. 2009).](image)
For a technical system to be self-organized it requires a decentralized control system where intelligent agents are able to make decisions with regards to how, when, and even if it should perform an activity, much in the same way as a social system. An approach to achieve this is to develop a distributed system, in which each agent is autonomous, locally interactive, and control is decentralized. Both holonic MS and evolvable MS have a distributed control structure which enables self-organization. However, the control structure of holonic MS is both heterarchical and hierarchical; i.e. the heterarchical structure embedded within *basic holons* is amended with a hierarchical control structure enforced by *staff holons* that give advice to *basic holons* (Van Brussel et al. 1998).

<table>
<thead>
<tr>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Central Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Distributed Control</td>
</tr>
<tr>
<td>Self-Organization</td>
<td>+++</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>(Operational Independence)</td>
</tr>
</tbody>
</table>

**Evolutionary Behavior**

*A guided “process of variation and natural selection of systems where selection is automatic and a result of the internal or external environment”* (Heylighen 1997b).

Evolution of a system is closely related to the life-cycle stages of that system, and if the system’s life-cycle is continuous or not. While continuous systems are required to evolve to answer to changes in its environment; noncontinuous systems are developed to function within a predetermined environment and therefore seldom possess the capability to evolve. However, systems that are unable to evolve beyond its predetermined requirements often exhibit some level of flexibility to cope with changes that are foreseeable and cost efficient to handle. Within production, the term *flexibility* is to some extent used as a synonym to *evolvability*; however, this relationship is only valid when a system’s flexibility extends the predefined changes to the internal and external environment, cf. discussion on classification of manufacturing flexibility (De Toni and Tonchia 1998).
Development of manufacturing systems is commonly executed through the classical system development stages. The purpose is to develop a system that meets specified requirements, and is stable, predictable, reliable, transparent, and controllable (Minai, Braha, and Y. Bar-Yam 2006). This result in a system that is unable to evolve based on internal and external changes in the environment, other than the predicted variations that are predefined in the requirement specification, i.e. flexibility.

The life-cycle of a distributed manufacturing system does not necessarily follow that of a centrally controlled manufacturing system. One of the purposes with a distributed manufacturing system is that its modules should be able to continuously negotiate their role in the system and be capable to evolve according to the circumstances to incorporate newly available technology, and thereby evolving the system based on immediate environmental needs (Barata et al. 2007). The system’s life-cycle is thereby detached from the life-cycle of its modules; i.e. changes to initial requirements, module failure, and outdated software and hardware do not necessarily result in system depletion or demise.

Both limited and extended production systems are logically continuous systems that evolve based on natural selection guided by its internal and external environment. Both the boundary and the internal properties of an evolutionary system are varying over time. Consequently, operation of an extended production system requires the capability to cope with evolutionary behavior.

<table>
<thead>
<tr>
<th>Evolutionary Behavior</th>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+++</td>
<td>++</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
</tbody>
</table>

48
Heterogeneity

Consisting of a wide variety of elements of different nature, dynamics and time scale, which requires knowledge from diverse scientific fields (Delaurentis 2005).

Given the hierarchical decomposition of production systems in this thesis, it is natural that an extended production system is more heterogeneous than a production system, which in turn more heterogeneous than a manufacturing system. Even so, all production systems are made up by a wide variety of “men, machines, methods, materials, and money” (Encyclopedia Britannica 2008); and these resources are simultaneously heterogeneous on multiple scales, e.g. size, architecture, life-cycle, scientific area, and elementary dynamics.

The environment and the functions required for a production system are however similar throughout different manufacturing industries. This enables development of standardized interfaces between different fields and interdisciplinary education programs, which reduces the problems related to heterogeneity.

<table>
<thead>
<tr>
<th></th>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneity</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Emergent Behavior

The patterns that result from interactions among elements in a system and their interactions with the environment (A. Clark 2000)

The issue of emergence in production systems is related to that there are multiple layers, e.g. a micro (local), meso (regional), and macro (global) level. Emergence occurs when elements on a lower level interact to create a behavior, or pattern, on a higher level.

A large part of reducible systems engineering is to reduce uncertainty, e.g. through modeling understand which meso and macro patterns will emerge. A complete understanding is unobtainable, which results in that engineering of complex systems is an iterative process that has to consider emergence as it
arises, and often results in a costly and time-consuming build-test-fix spiral. A production example of this is the deployment of a manufacturing system, where the integration far exceeds the design of the mechanical system Figure 21 (EUPASS 2004). The figure could be interpreted to mean that a non-modular centralized system exhibits more emergent behavior, which is true in the sense that more unexpected patterns arise for each new product. However, the key issue here is the ability to handle and take advantage of emergence, rather than the inability to foresee them.

Figure 21: Design and integration of a manufacturing system for a new product, comparison between a non-modular centralized system and an evolvable production system (Neves and Onori 2009).

Behaviors that emerge through homogeneous system states are naturally easier to predict than the heterogeneous; consequently, it is more difficult to predict emergent behaviors for an extended production system than for a regular production system. Also, at higher levels of abstraction, the boundary conditions associated with the internal and external environment becomes less clear, leading to uncertainties in modeling and simulation.

<table>
<thead>
<tr>
<th></th>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Emergent Behavior</td>
<td>+++</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>
Complex Networks (Small-World and Scale-Free)

*Exhibiting statistical properties “common to a variety of diverse real-world social, information, biological, and technological networks”* (Braha and Y. Bar-Yam 2004a).

*Small-world: most nodes are not neighbors, but can be reached from every other node in few steps.*

*Scale-free: the degree distribution follows a power-law, i.e. there are very few highly connected nodes (hubs) and many nodes with few connections.*

Small-world networks and scale-free networks are properties that were initially discovered in a wide variety of large-scale systems, see e.g. (Watts and Strogatz 1998). Common to all are that network data is available or easily acquired, and that the data is unambiguous. For heterogeneous systems the network analyst must first determine the types of nodes and edges to be included in the network, only then can data gathering be initiated.

Network analysis of production systems have mainly been done on supply networks, i.e. extended production systems; and on engineering problem solving networks, i.e. activities within a production system aimed to develop a complex product or a manufacturing system.

In a supply network, the members are engaged in shared activities to fulfill the goals of the customers (D. Simchi-Levi, Kaminsky, and E. Simchi-Levi 2000). However, since rewards may differ between different nodes and each member is autonomous; all members make independent and local decisions to maximize the likelihood of achieving the goal (Pathak, Dilts, and Biswas 2007). A supply network consists of: an environment (E), its nodes \( n_i \), and relations (R). According to Pathak et al, these three are defined through a set of components: \( E = ( \text{operational rules, fitness thresholds, environmental demand, price, and number of nodes (the latter three varying over time)}) \); \( n_i = ( \text{objectives, constraints, strategies, learning ability, and fitness value}) \); and \( R = ( \text{nodes, and edges; which generates a bidirectional graph}) \). The state of each of these components affects the topology of the network, whereby supply network topologies evolve differently in different industries. For example, in the US automotive industry,
the supply network is composed of “a few major assembly plants, many direct suppliers, and a multitude of lower-tiered suppliers”, and can be characterized as “deeply structured, hierarchical, and consolidated” (Pathak, Dilts, and Biswas 2007). In other industries that exhibit different component states other topologies evolve, e.g. defense suppliers, natural resource industries.

In a network analysis of the supplier-prime buyer relationships in the Japanese interfirm network Ohta, it is shown that the supply network neither exhibit small-world nor scale-free behavior (Nakano and D.R. White 2006). In this study over 7000 SMEs were surveyed about their three prime buyers, which generated a supplier-prime buyer network of over 8000 interrelated companies. The network illustration in Figure 22 shows both the number of suppliers for each company (size of spheres), and their hierarchical location in the network (flow of material going upwards). The companies were mainly SMEs, but some top original equipment manufacturers were also part of the network, e.g. Toshiba (112 suppliers), NEC (53), and Hitachi (45).

Figure 22: Global structure of Ohta supplier-prime buyer network (Nakano and D.R. White 2006).
The Ohta network exhibited some of the required network properties; however, it failed as a small-world model mainly because the lack of local clustering, i.e. the likelihood that two nodes that are connected to a third are also connected to each other. In the Ohta case this was particularly clear at the higher supplier levels, where two companies that both were connected to one specific supplier seldom collaborated between themselves. The scale-free network model was also rejected because the network was primarily organized by the most connected companies, i.e. the hubs in the network. This is not congruent with scale-free behavior, which is based on a bottom-up organization of the network.

Explanation for why manufacturing supplier networks exhibit neither small-world nor scale-free behavior possibly lies in that (1) the supply network is often modeled after the actual product structure (Eppinger et al. 1994), which due to mechanical and geometrical constraints has a low clustering coefficient and nodal degree (Whitney 2003); (2) the supply network is one-dimensional and shows only the top three buyers for each company, this approach will not reveal the less traditional links between different companies that a multi-dimensional analysis would.

Similar to supply networks, the topology of complex product development (PD) networks vary depending on multiple components within the network and the system to be developed, e.g. system size, complexity, organization, industry. However, a wide variety of large-scale product development networks have show to be both small-world and scale-free networks, e.g. vehicle development, operating software development, pharmaceutical facility development, and hospital facility development (Braha and Y. Bar-Yam 2004a), jet-engine development (Bradley and Yassine 2006). These types of engineering problems are similar to production systems engineering in size and complexity. For the engineering problem-solving networks above the number of people involved (nodes) range between 120 and 889, in the jet-engine development case there were 54 interrelated design teams. The number of people involved in production system development is naturally different for differently sized systems and organizations; however, an example of size is given by a
BMW engine factory case, where 70 business processes were to be designed and implemented, totaling approximately 200 people involved throughout the project (Wu et al. 2000).

The network properties of the engineering development networks presented by Braha and Bar-Yam are presented in Table 6 together with the network properties of the Ohta supply chain. Conclusions that can be drawn from these properties are: (1) PD networks are similar to other real-world networks; (2) PD networks react quickly to design changes (short path length); (3) PD is centered on a few key tasks, i.e. hubs; (4) efforts should be focused on improvement of key tasks; (5) failure of key tasks is devastating for the whole PD; (6) removing some links could reduce the time and cost of the PD, but it could also reduce the quality of the PD and the product (Braha and Y. Bar-Yam 2004b).

Table 6: Network measures for product development networks (*) (Braha and Y. Bar-Yam 2004a) and a supply network (†) (Nakano and D.R. White 2006).

<table>
<thead>
<tr>
<th>Network</th>
<th>N (number of nodes)</th>
<th>K (average degree)</th>
<th>L (actual)</th>
<th>L (random)</th>
<th>C (actual)</th>
<th>C (random)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Design (*)</td>
<td>120</td>
<td>6,95</td>
<td>2,878</td>
<td>2,698</td>
<td>0,205</td>
<td>0,07</td>
</tr>
<tr>
<td>Operating Software Design (*)</td>
<td>466</td>
<td>5,34</td>
<td>3,7</td>
<td>3,448</td>
<td>0,327</td>
<td>0,021</td>
</tr>
<tr>
<td>Pharmaceutical Facility Design (*)</td>
<td>582</td>
<td>14,17</td>
<td>2,628</td>
<td>2,771</td>
<td>0,449</td>
<td>0,023</td>
</tr>
<tr>
<td>Hospital Facility (*)</td>
<td>889</td>
<td>18,40</td>
<td>3,118</td>
<td>2,583</td>
<td>0,274</td>
<td>0,024</td>
</tr>
<tr>
<td>Ohta interfirm networks (†)</td>
<td>8347</td>
<td>2,3849</td>
<td>8,86</td>
<td>9,77</td>
<td>0,000052</td>
<td>0,00053</td>
</tr>
</tbody>
</table>

These small-world and scale-free engineering development networks exist within both extended and limited production system. Even though the organization of production systems is often hierarchically structured, they possess the ability to form small-world and scale-free networks.

Manufacturing systems are comparable to other technical systems, and their network properties can be evaluated based on system
components deliberately connected to each other. Unlike in a social or biological system, a connected node in a technical system is often constantly occupying a percentage of another node’s connecting capability. An assembly that is over-constrained is sensitive to parts’ variation, is difficult to assemble, and is operationally poor. The fundamental constraints impacting an assembly/component network result in that it does not necessarily exhibit the same characteristics as a social or biological network, and that a mechanical assembly is rarely scale-free (Whitney 2003).

This technical view is comparable to a snapshot view of an evolvable manufacturing system. Long-term network characteristics, e.g. the relationships between modules over time, have not been addressed, and are difficult estimate.

<table>
<thead>
<tr>
<th></th>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Central Control</td>
</tr>
<tr>
<td>Scale-Free/Small-World</td>
<td>++</td>
<td>+++</td>
<td>-</td>
</tr>
</tbody>
</table>

### 3.1.2 Discussion of Hypothesis A

To provide a clear overview, the results from the previous section are presented in Table 7. Each of the 20 predicates has been tested and evaluated.

Table 7: SoS characteristics for different production systems.

<table>
<thead>
<tr>
<th></th>
<th>Extended Production System</th>
<th>Production System</th>
<th>Manufacturing System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Central Control</td>
</tr>
<tr>
<td>Self-Organization</td>
<td>+++</td>
<td>+</td>
<td>–</td>
</tr>
<tr>
<td>(Operational Independence)</td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Evolutionary Behavior</td>
<td>+++</td>
<td>++</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Emergent Behavior</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Scale-Free/Small-World</td>
<td>++</td>
<td>+++</td>
<td>–</td>
</tr>
<tr>
<td>(not necessarily organizational)</td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
From the table above, it is clear that both types of production systems exhibit most SoS characteristics. For all characteristics except scale-free and small-world, the extended production systems display a stronger degree.

Centrally controlled manufacturing systems exhibit only a few SoS characteristics and those to a limited degree, in addition to lacking much of the societal complexity, it cannot be considered a SoS. The human/societal complexity of a distributed manufacturing systems is also too small to be considered a SoS, however, the multi-agent approach with intelligent modules does in many ways qualify evolvable production systems as having enough societal complexity to be a SoS. In addition, distributed manufacturing systems exhibit some degree of most SoS characteristics.

It is important to understand that it is each individual predicate that has been tested, not the set of predicates for each production system type. With this in mind; based on the definitions of both SoS and different types of production systems, the hypothesis can be corroborated with regards to extended production systems, partially corroborated for regular production systems and for distributed manufacturing systems (due to a limited degrees of most SoS characteristic), and falsified for centrally controlled manufacturing systems (due to lack of both societal complexity and SoS characteristics).
3.2 Hypothesis B

Limitations to an individual’s capability to conceptualize disable the ability to successfully design SoS with methods for reducible systems engineering.

Hypothesis B states that when reducible systems engineering is used for the engineering of SoS issues it requires engineering abilities that no individual possesses. More specifically, RSE is structured so that one individual is responsible for the whole system, something that is achievable because a RS can be decomposed without loss of any vital information. If the same engineering structure is used for SoS it necessitates that one individual is able to fully conceptualize the whole un-decomposed system, which is not achievable because SoS cannot be decomposed into separate modules without losing key information.

Hypothesis B is tested in two stages: (1) a framework for modeling the contents of an individual’s conceptualization and two key limitations (chapter 2.3); (2) the two key limitations of conceptualizing, information content and scale, are individually related to each SoS characteristic. The results are then discussed with regards to falsification or corroboration the hypothesis.

3.2.1 Effects of the Limitations to Conceptualizing

The process of conceptualizing has been described in chapter 2.3, and two key limitations related to the capacity and capability of an individual were established:

The capacity of the human brain and sensory organs are finite, which leads to that the observer is only able to conceptualize a limited part of the reality at the time, and must make an adjustment to be able to conceptualize another part of the reality.

An observer is only able to conceptualize at one scale at the time, which leads to that an observer is not able to hold two diverse perspectives at the same time, and is thereby not able to compare multiple perspectives without changing the conceptualization back and forth.
An individual has a limited capacity to store and process information; consequently, the information content of the relevant level of the studied system must be less than the capacity of the individual in order for the system to be fully understood at once (Figure 23).

![Figure 23: A system must be decomposed in accordance with an individual's ability to handle information.](image)

An individual can only conceptualize one scale at the time; consequently, the information content of the relevant level of the studied system must be of the same scale in order for the system to be fully understood at once (Figure 24). Multiple scales may lead to a gap between the capacity of an individual and the information content at a specific scale, which enables the individual to also focus on negotiating with interrelated systems in accordance with SoSE.

![Figure 24: A system must be decomposed in accordance with an individual's ability to handle only one scale at the same time.](image)

A key consequence for SoS engineering is that a requirement specification cannot be fully described at one time for a SoS, which leads to that a SoS cannot be decomposed into subsystems with clear boundaries. The requirements specification must therefore be dynamic and grown over time, and the subsystems of a SoS must continuously negotiate their internal and external interfaces.
In the following sections, each SoS characteristic is individually correlated with the capacity and capability of an individual, which is summarized in a fractional table; these tables are brought together and further analyzed in Chapter 3.2.2, and are therefore not numbered individually.

**Self-Organization**

“*a process where order emerges without external control based on local interactions of constituent components*” (NECSI Wiki 2008a).

A reducible system has clear boundaries, a purpose, and can in the preliminary stages be decomposed into subunits, without loss of any essential information. This enables the possibility for the system to be centrally controlled, which often precludes self-organization. Given that SoS are non-reducible, a SoS cannot be decomposed without loss of vital information. As a result, the whole system must be conceptualized by one individual, something that is impossible both regarding capacity and capability.

The issue of insufficient capacity to conceptualize a system becomes obvious. The information content of a SoS greatly exceeds an individual’s abilities, and a hierarchical decomposition therefore becomes impossible. A self-organized system is instead engineered bottom-up, which allows for allocation of a manageable amount of information content to each individual.

The issue of multiple scales is also important for the emergence of self-organizing systems. Since an individual is unable to simultaneously conceptualize multiple scales, a non-reducible system should be organized so that only one scale is required to be conceptualized at one time by one individual. Engineering of a self-organizational system handles multiple scales through low-level negotiation of the variables that are of interest between the individual organizational units.

<table>
<thead>
<tr>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Organization</td>
<td>Direct and strong cause</td>
</tr>
</tbody>
</table>
Evolutionary Behavior

A guided “process of variation and natural selection of systems where selection is automatic and a result of the internal or external environment” (Heylighen 1997b).

The environment of a socio-technical system is constantly changing, SoS must therefore possess the ability to alter its current state to a state that is competitive in its new environment. This ability can be understood as the system’s ability to remain fit. A common strategy to handle a changing environment is to provide the system with some measure of flexibility, so that the system is able to handle some changes to the environment. This strategy becomes impractical when environmental changes are large or unpredictable, and also because flexibility implies that the system is adequate in performing a range of activities, but never performs one activity optimally, which drives cost and decreases reliability. For SoS, the main strategy is therefore to enable evolutionary behavior, i.e. that the system is able to reconfigure itself into a state that correlates with the environment.

To be able to conceptualize an evolving system, a new focus of attention must continuously be reestablished. This is achieved through simultaneous observation and conceptualization of the dynamic system and its dynamic environment. As with self-organization, this requires processing of large amounts of information over multiple scales, and is therefore not feasible for one observer. As a result, if the observer is unable to realign the conceptualization or is unaware of the evolution itself, the conceptualization will not be corresponding with the reality.

The main issue in addressing an evolutionary system with RSE is that the system and its environment are assumed to be stable and predictable (Minai, Braha, and Y. Bar-Yam 2006), and that one solution will solve the stated problem for a certain period of time. Since SoS are neither stable nor predictable, a reducible systems engineering approach will not be successful.

<table>
<thead>
<tr>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolutionary Behavior</td>
<td>Indirect and contributing</td>
</tr>
</tbody>
</table>
Heterogeneity

*Consisting of a wide variety of elements of different nature, dynamics and time scale, which requires knowledge from diverse scientific fields* (Delaurentis 2005).

A heterogeneous system has per definition multiple scales. The level of detail and abstraction has a strong affect on which scientific areas are of relevance for a specific issue. An issue in conceptualizing a heterogeneous system is that the identification and labeling of patterns are more difficult since it is not certain that two neighboring patterns belong to the same area of knowledge. This requires that the observer is able to absorb and process more information for a heterogeneous than a homogeneous system. In a heterogeneous system it is also necessary to have a wider range of *focus of attention, field of view, and resolution* in order to detect and distinguish patterns in the system, which is a capacity issue rather than capability.

Heterogeneity leads to an increase of a system’s information content and number of scales. Since this becomes impossible to handle in a centrally controlled non-reducible system, a self-organizing approach is required.

<table>
<thead>
<tr>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneity</td>
<td>Strongly affected but not cause.</td>
</tr>
</tbody>
</table>

Emergent Behavior

*The patterns that result from interactions among elements in a system and their interactions with the environment* (A. Clark 2000)

Emergent behavior is a consequence of the discrepancy between the reality and the conceptualization, and is therefore a direct result of both the capacity and capability of an individual.

In a reducible system, emergence can be avoided by understanding the couplings between subsystems in a top-down decomposition. The system is then decomposed into units that answer to an individual’s capacity and capability. In a SoS, subsystem couplings cannot be fully understood by an individual,
i.e. the information content of the system exceeds the information handling ability of the observer, it is therefore necessary to develop the system bottom-up. Emergence is consequently a driver of self-organization.

<table>
<thead>
<tr>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergence</td>
<td>Direct and strong cause</td>
</tr>
</tbody>
</table>

Complex Networks (Small-World and Scale-Free)

*Exhibiting statistical properties “common to a variety of diverse real-world social, information, biological, and technological networks”* (Braha and Y. Bar-Yam 2004a).

*Small-world: most nodes are not neighbors, but can be reached from every other node in few steps.*

*Scale-free: the degree distribution follows a power-law, i.e. there are very few highly connected nodes (hubs) and many nodes with few connections.*

The occurrence of network properties is dependent on the complexity of the system that is being engineered and the capacity of each node. It could therefore be argued that there is a causal link between network properties and capacity to conceptualize.

If an individual is not able to connect to enough nodes itself, the network tries to develop characteristics that enable the individual to achieve a similar result, which may lead to a small-world network. The limitations to capacity and capability lead to that different nodes focus on different issues, some focus on connecting the network (hubs), while others are not focusing on connecting the network, leading to a scale-free network.

It is difficult to determine any direct relationship between the capability to conceptualize only one scale and the development of small-world and scale-free behavior. However, systems are commonly able to bridge inabilities at the nodal level through the development of networks.

<table>
<thead>
<tr>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small-World and Scale-Free</td>
<td>Direct and strong cause</td>
</tr>
</tbody>
</table>

| Uncertain                      |
3.2.2 Discussion of Hypothesis B

The ability to conceptualize is essential during the whole lifecycle of a system. It is therefore necessary to understand the building blocks of a conceptualization, and how these are affected by the person forming it. The presented framework has provided a foundation to advance the knowledge of what a conceptualization is and the limitations to an individual’s abilities.

The correlation between SoS characteristics and an individual’s capacity to handle information content and the inability to conceptualize more than one scale at the time is evaluated and presented in the five by two matrix below (Table 8). Each predicate is discussed on the grounds of if the conceptualization ability contributes to the SoS characteristic and if it even is the cause of it. It is also discussed whether there is a direct or indirect link between the columns and the rows.

Table 8: Evaluation of correlation between SoS characteristics and the capacity and capability of an individual.

<table>
<thead>
<tr>
<th></th>
<th>Information Content (Capacity)</th>
<th>Multiple Scales (Capability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Organization</td>
<td>Direct and strong cause</td>
<td>Direct and strong cause</td>
</tr>
<tr>
<td>Evolutionary Behavior</td>
<td>Indirect and contributing</td>
<td>Indirect and contributing</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>Strongly affected but not cause.</td>
<td>Direct and strong cause</td>
</tr>
<tr>
<td>Emergent Behavior</td>
<td>Direct and strong cause</td>
<td>Direct and strong cause</td>
</tr>
<tr>
<td>Scale-Free/Small-World</td>
<td>Direct and strong cause</td>
<td>Uncertain</td>
</tr>
</tbody>
</table>

For all generic SoS properties there is a link to the capacity and the capability, with the exception of the relations between the inability to conceptualize more than one scale and scale-free/small-world, the origin of which has not been fully explored.

There is a direct and strong relationship to both self-organization and emergent behavior; the former becomes evident for non-reducible systems; and the latter is evident since emergence often can be understood as the discrepancy between the reality and an individual’s static and dynamic perception of the reality.
The *evolutionary behavior* of a system is a positive result that comes from it being *self-organized*, and is therefore only indirectly affected by an individual’s abilities.

The reason that *heterogeneity* is a characteristic of SoS is directly related to our inability to conceptualize multiple scales, if that was not the case it would only be the information content that was of importance.

The development of a system is strongly dependent on the abilities of the nodes, e.g. the number of connections that each node is capable to have. A clear link can therefore be seen to both the capacity and capability of an individual; however, network analysis is usually one-dimensional and it is therefore difficult to find information on how a network is affected by our inability to conceptualize multiple scales. This means that RSE becomes insufficient even if only one SoS characteristic is present in a system. The key issue is not whether a system is a SoS, but if the system can be sufficiently engineered with reducible systems engineering methods.

The strongest relationship between SoS characteristics and the capacity and capability is found for *emergent behavior*. This is understandable since emergent behavior to some extent can be foreseen through extensive simulation and modeling.

Another key result is that *evolutionary behavior*, *heterogeneity*, and *emergent behavior* are all drivers of *self-organization*. Each of these properties requires a capability and capacity to conceptualize that is unmanageable through reducible systems engineering. The only feasible solution is to allow for *self-organization*, to build the system bottom-up and thereby engineering it in accordance with the abilities of each individual.

It has been shown that it is not feasible to address SoS characteristics with reducible systems engineering. The rationale for this lies in that RSE requires superhuman capability and capacity to conceptualize a system. The conclusion is therefore that hypothesis B has been corroborated.

The inability to conceptualize infinite amounts of information at multiple scales leads to that each individual is only able to handle a
limited part of a system. The decomposition of a reducible system is controlled by the information content of each subsystem, the scale, and by the information content of the interfaces between subsystems (Figure 25). The purpose of RSE is then to remove uncertainties in the architecture and each subsystem.

Figure 25: Decomposition of a reducible system.

The inevitable loss of information during SoS decomposition requires a different, non hierarchical approach to engineering of SoS. There is a limited knowledge of the purpose of a SoS and how to achieve them in the initial stages. The only possibility is therefore to initiate development on a lower system level, and then allowing the system to grow through self-organization (Figure 26). These development nodes will grow in accordance with the internal fitness, which is dependent on the interface to other development nodes and to the SoS environment. Each node can grow until the information content becomes too large for one person to conceptualize. At that point the node should split into two or more self-organizing nodes.

Figure 26: Evolution of a SoS; starts with an intention, is initiated locally and grows in accordance with internal and external relationships.
4 Final Discussion and Conclusion

This chapter provides a critical review of the presented research, suggestions for future research, and a concluding summary of the results presented in this thesis.

4.1 Critical Review

In this thesis, the existence of non-traditional system properties in different types of production systems has been explored. It has been shown that properties that are used to define System of Systems are also present in production systems. A link between the areas of SoS research and Production system research has thereby been established, and a new area of interdisciplinary research has been initiated.

Based on previous research within SoS and complex systems, important implications of the occurrence of SoS properties have been addressed and interrelated with engineering of production systems. The understanding of how SoS properties can be addressed has been further expanded using a conceptualization framework that explains how an individual conceptualizes a system. This approach has provided an understanding of when reducible (traditional) systems engineering is appropriate, and when another approach is required and why.

The nature of the research and the research area also give rise to some negative criticism. First of all it needs to be pointed out that the purpose of this thesis is neither to provide a method for detecting SoS properties in a specific production system, nor a method for addressing SoS properties in engineering. The thesis is rather offering a framework that can help production engineers and managers to understanding that SoS properties may exist in their system, and that they require a different approach.

The research area has a very broad scope, is inherently multidisciplinary, and barely researched. While this signals that much can be done, it also means that there are no established approaches that are tuned to production systems and data relevant for quantitative analyses are scares. A relevant question is therefore if the hypotheses really can be corroborated or falsified?
For hypothesis A, each SoS characteristic is validated through a variety of production systems; however, all SoS characteristics have not been validated for one specific production system instantiation, i.e. the coincident occurrence of all SoS characteristics in one production system has not been tested. Consequently, it has not been shown whether production systems can be SoS, only that some of them exhibit SoS characteristics. However, the assessment of hypothesis B showed that each individual SoS property requires a different approach than RSE to be addressed. Consequently, it does not matter much if a system exhibit one or all of the SoS properties (with the possible exception of small-world and scale-free), what matters is that at least one property is exhibited. In addition, it was also shown that evolutionary behavior, heterogeneity, and emergent behavior are all drivers of self-organization, which means that if one of them is found in a SoS, there is an increased chance that the system also exhibits self-organization.

The scientific sources that the results in this thesis are based on address a variety of different types of production systems and even systems that resemble production systems. Sources were selected based on the complexity of the studied systems: primarily sources on production systems for complex products, e.g. automotive, aeronautic, were chosen; secondarily, sources on production of less complex products and engineering of non-production related systems. However, the purpose has been to establish that some production systems exhibit SoS properties, not that all production systems always exhibit SoS properties.

The conceptualization framework is used to explain how an individual make sense of the real world, and aims to clarify the limitations of an individual. The framework does not; however, explain how the human brain actually works. That is beyond the scope of this thesis.
4.2 Future Research

The application of SoS and complex system research to production is essentially unexplored. The possibilities for future research are therefore many. Some of the more interesting ideas include:

Further validation of the hypotheses through case studies. These could include exploring SoS properties in a specific production system; determining if and how they are currently addressed; and determining how a production system can benefit from SoS properties.

Evolvable Production Systems (EPS) and Evolvable Assembly Systems (EAS) are two very interesting approaches that incorporate several of the SoS properties. These systems would be ideal test beds since they are limited in size and complexity in comparison with most SoS. The engineering of these systems could be further developed in line with SoSE. It would also be of great interest to explore how initial rules of operation in a self-organizing system affect the network structure of that system.

Systems Engineering is a well-established set of tools and methods that is very useful for reducible systems; however, it would be of great interest to explore the possibilities to merge SoSE ideas into systems engineering to obtain additional advantages.

The area of SoS is not matured and would benefit from interdisciplinary collaborations between all areas working within complex systems, e.g. biology, social sciences, infrastructure, military research, production systems. Such an approach could greatly advance all participating research areas as well as complex systems research. While there are a few international collaborations, a national collaboration on complex systems could provide valuable opportunities for interdisciplinary collaborations and increase the benefits of a complex systems approach.
4.3 Conclusions

Research on system of systems and complex systems has become increasingly popular over the last decades, and more areas are incorporated and labeled complex. A key reason for this is that technological advancements allow more data to be captured and processed, which both drives complexity and analysis of complexity.

This research attempts to contribute to scientific body of knowledge within the research areas of both production systems and complex systems; on one hand by establishing that SoS characteristics are also characteristics of production systems; and on the other hand by establishing that an individual’s ability to conceptualize a system affects both the system itself and how it can be engineered.

It has been shown that extended production systems can exhibit all SoS characteristics to a high degree; that production systems and distributed manufacturing systems exhibit most SoS characteristics, usually to a slightly lower degree than extended production systems; and that centrally controlled barely exhibit two SoS characteristics.

It has been shown that an individual’s ability to conceptualize a system greatly affects the process of engineering. While the current systems engineering process is suitable and sufficient for reducible systems, it is not possible to successfully engineer a SoS with reducible systems engineering. The main reason for this lies in that a SoS cannot be decomposed without loss of vital information. The solution is therefore to develop an engineering framework that supports evolution through self-organization.
References


