Communications of the IIMA

Volume 16 | Issue 3 Article 1

2018

Identifying Relationships of Interest in Complex Environments by Using Channel Theory

Andreas Bildstein
Fraunhofer IPA, andreas.bildstein@ipa.fraunhofer.de

Junkang Feng
School of Engineering and Computing of UWS, junkang.feng@uws.ac.uk

Identifying Relationships of Interest in Complex Environments by Using Channel Theory

As requested in the submission process, this paper is without title page (incl. author's names, abstract, keywords).

INTRODUCTION

Complexity increases in almost all areas of our daily lives and being able to deal with challenges arising from this increasing complexity is important. In this paper, we describe how we can handle complexity in complex environments, which can be seen as distributed systems as they are described in the theory of Information Flow (Barwise & Seligman, 1997), also known as Channel Theory. In this kind of distributed systems, two or more sets of objects are each organised in so-called *contexts* and the objects are standing in some relationships to each other.

To understand what can be done to handle the complexity in those kinds of environments, we must understand what complexity in general means and which kind of properties a complex system shows. Cilliers (2002) gives a description of the characteristics of *complex systems* and explains the important distinction between systems that are complex and those systems that are complicated. According to Cilliers, with the help of complete knowledge about all the components of a system, a complicated system can be described and understood completely. In contrast, analysing and having deep knowledge about all the components of a complex system is not enough to describe and understand such kind of systems completely as the interactions and relationships between the overall system and its components have to be taken into account as well to get a decent understanding of the whole system. In this work, we adopt Cilliers' definition of complex systems for our work in the context of distributed systems.

By analysing the relationships between objects in complex environments, we are able to learn about the regularities that govern the interactions between the objects within complex environments. If we have insights on some of the regularities between objects within a system, we can better predict the system's behaviour under specific circumstances. The more we know about the regularities and the relationships within a system, the better we can predict the behaviour of the system as we are taming the level of uncertainty within this system—and uncertainty within in a system is one of the main drivers of complexity within this system.

To find the relationships and thus the regularities within a complex system, this paper introduces a solution approach based on Channel Theory, a well-established theory from information science and semantic information theories—more details on Channel Theory, its main components and its usage will be given in the next section. The proposed approach in this paper enhances the traditional application of Channel Theory with an iterative procedure for being able to identify relationships of interest even in complex environments. Such a complex environment can be found in the application domain of smart manufacturing, which we choose as the exemplary application scenario in this paper.

One of the main theories behind the worldwide 'Factory of the Future' movement (also known as smart manufacturing in the US or Industrie 4.0 in Germany) is the idea that products, production equipment, and production IT systems are getting far more interconnected as they are at the moment to finally reach a certain level of self-organization and autonomy. An example of this self-organization in smart manufacturing is a production system, which decides without human intervention on which production equipment should be used to conduct a specific production step.

Caused by globalisation and customer requirements for more customised and even personalised products, many manufacturing companies have to react to an increasing number of product variants while the number of sold products per product variant declines. This results in an increasing need for flexible production systems (Koren, 2010), which are able to handle a rising variety of different combinations of equipment and tools able to conduct a growing number of different and new products and thus production steps. To manage this increasing complexity in the decision on which equipment should be used to conduct a specific production step an automated equipment assignment solution might be desirable.

Within our use case example for this paper we investigate how the assignment of production resources can be supported even in a production environment with an increasing number of known and also unknown products by applying Channel Theory. The application scenario starts where the necessary production steps and their specifications like material, dimensions or surface and tolerance requirements are already defined and now have to be conducted by suitable production equipment. Thus, we focus on the question of what production equipment is capable of conducting a specific production step like making a hole of a specific diameter in a work piece with a particular thickness and a given material.

The alignment process of our application scenario shall reveal which equipment can be used to conduct a specific production step based on a matching between the production step specifications and the capabilities description of the available equipment. The matching between product specifications and production equipment capabilities is a quite generic task. Thus, our application scenario might not only be of interest for a local production environment within a single factory, but also for establishing dynamic manufacturing networks (Papakostas, Efthymiou, Georgoulias, & Chryssolouris, 2012) with the help of cloud-based infrastructure (Stock & Bildstein, 2015) or within the emerging business concept of cloud manufacturing (W. Li and J. Mehnen (eds.), 2013).

RELATED RESEARCH

Channel Theory

To identify the relationships between the objects within a distributed and complex system, we use the theory of Information Flow (IF for short), also known as Channel Theory, put forward by Barwise and Seligman in 1997. We are using Channel Theory as this theory provides us with the mathematical tools that help us to describe the flow of information within a distributed system and thus to find out about the regularities within such a system. Furthermore, this theory has been successfully applied in a series of different scenarios where the relationships of two or more sets of things have to be determined. In those application scenarios, the relationships are often of different kinds, but all those relationships have in common that they base upon the flow of information between two or more different sets of things within a distributed system and thus have an informational origin—they are information-based relationships. And this search for information-based relationships in distributed systems distinguishes this IF-based approach from other approaches in the broad field of computer science, e.g. in the field of Artificial Intelligence and Machine Learning, which provides us with different methodologies and approaches, aiming to determine relationships, links, or associations between separate sets of things as well.

Examples of the successful application of Channel Theory are the work of Kalfoglou and Schorlemmer, who developed IF-Map (Kalfoglou & Schorlemmer, 2003) as an IF-based methodology for the mapping of ontologies. Over the years, Kalfoglou and Schorlemmer refined their ideas on the application of Information Flow and published additional work in different application areas on how Information Flow might be used. They used it especially in scenarios where ontologies have to be aligned semantically, e.g. see Schorlemmer & Kalfoglou (2003a), Kalfoglou & Schorlemmer (2005), Schorlemmer & Kalfoglou (2005), Schorlemmer & Kalfoglou (2007), and Kalfoglou & Schorlemmer (2010).

Based on this work, successive researchers applied Kalfoglou and Schorlemmer's approach in a series of notional and real-world problems, at which they applied or even modified the methodology they have seen in the work from Kalfoglou and Schorlemmer. Amongst those researchers of the second generation, we can find, for example, Xu and Feng (2012), who show an example of two questionnaires and a small set of questions within those questionnaires that shall be integrated. Also, the work of Yang and Feng (2012) stands in this tradition and shows a scenario where two databases with employees from two merged companies shall be integrated based on their locations. We can find similar implementations for example in the work of Mellal and Dapoiny (2007), Wang and Feng (2007), Mantri (2013), and Yang (2015).

Main components of Channel Theory

The main components for the application of Channel Theory are classifications, infomorphisms, correspondences and constraints. Those components are used to construct an IF channel and will be introduced within this section.

Channel Theory: classifications

In Channel Theory, a component of a distributed system is modelled with the help of a mathematical structure called classification. This classification represents the context of this component and consists of particulars (objects) and attributes that help to describe those particulars. In Channel Theory, the particulars, e.g. production step 1, are named tokens and the attributes that classify those tokens, e.g. diameter=12mm or material=stainless steel, are named types.

By classifying the tokens with types, a relation between the particulars and the attributes within a component is given. Definition of a classification (Barwise & Seligman, 1997; p.69):

> "A classification A = $\langle tok(A), typ(A), \models_A \rangle$ consists of a set, tok(A), of objects to be classified, called the tokens of A, a set, typ(A), of objects used to classify the tokens, called the types of A, and a binary relation, $\models_A$, between tok(A) and typ(A)."

We depict a classification in Fig. 1:

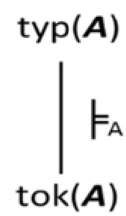

Fig. 1: Classification A

So, within a classification A, the binary relation classifies the tokens a_i of A to the types α_i of A in the form that the binary relation $\models_A$ is a subset of the Cartesian product between the types and the tokens of A, $\models_A \subseteq tok(A) \times typ(A)$.

Such a classification might be used, for example, to build up a context $ET_{Drilling}$, which consists of a set of different drilling machines $tok(ET_{Drilling})$ and a set of production capabilities $typ(ET_{Drilling})$, which those drilling machines provide.

Channel Theory: infomorphisms

In Channel Theory, classifications are connected with each other via so-called infomorphisms. The definition of an infomorphism can been found in (Barwise & Seligman, 1997; p.72):

"An infomorphism f: A ⇄ B from A to B consists of two classifications $\langle A, B\rangle$ and a contravariant pair $f = \langle f\hat{}, f\check{}\rangle$ of functions between A and B, satisfying the following fundamental property of infomorphisms:

$$f\check{}(b) \models_A \alpha \quad \text{iff} \quad b \models_B f\hat{}(\alpha)$$

for each token b ∊ tok(B) and each type α ∊ typ(A)."

With the help of these infomorphisms, information can be carried back and forth between the component classifications. The information that is being carried is the fact that a specific token a is classified to a specific type α, meaning the information that a is being of type α. We can depict an infomorphism as shown in Fig. 2:

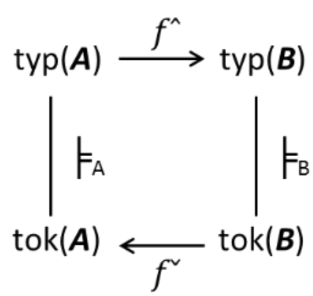

Fig. 2: Infomorphism f: A ⇄ B

Between a component and the system as a whole, there is at least one such infomorphism. An infomorphism is a pair of functions each of which captures correspondences between types or between tokens of two classifications that comply with the above fundamental property. Moreover, through the system as a whole (which would be represented as the 'core' of the information channel, which will be discussed shortly), relationships between components are also captured.

With the help of such infomorphisms, relationships between the tokens or between the types of two components of an information channel are represented. The function on the token-level of an infomorphism, for example, can represent the information that a specific detailed described production step $x \in tok(DD_{makingHole})$, which is of type making hole, can be conducted with a machine $y \in tok(ET_{Drilling})$, which is of type drilling machine.

Channel theory: correspondences

We have shown that an infomorphism is a pair of functions, which respectively represent the correspondences between the types and between the tokens of classifications within a distributed system. The correspondences themselves represent relationships between types or between tokens of the involved classifications.

As such, so far known correspondences are an important starting point when we have to match between the different components of a distributed system. Those initial correspondences are the result of a priori knowledge or other kinds of heuristics but can be also the result of a posteriori knowledge when we feed back our experience to this initial partial alignment. Thus, correspondences give us the information, what we already know about the relationships between the tokens or between the types of two contexts. In our application scenario, this might be, for example, the knowledge that the production step x of type $PS_{makingHole}$ can be conducted in the setting of a specific production environment with the production processes PP = {drilling, milling, turning, punching}.

In Information Flow, a correspondence is a pair of elements, which contains either two tokens or two types from the corresponding IF classifications and describes a particular relationship between the two component classifications. This pair of elements is then be used to build up a token $t_x = <t_i,A_x;t_j,B_x>$ within the core classification of the IF channel and this token t_x is described by the types of the involved tokens from the IF classifications A and B that build up this token t_x.

Channel theory: constraints

According to the first principle of information flow (Barwise & Seligman, 1997; p.8), the flow of information heavily depends on regularities in the distributed system. The more random a distributed system is the less information is able to flow between the components of this distributed system. Thus, the aim is to find as much regularities as possible in a distributed system to reach a stable alignment framework between the different components, namely classifications, within the distributed system. In Channel Theory those regularities are called constraints and are defined as follows (Barwise & Seligman, 1997, p.29):

> "Let A be a classification and let $\langle \Gamma | \Delta \rangle$ be a sequent of A. A token a of A satisfies $\langle \Gamma | \Delta \rangle$ provided that if a is of type α for every $\alpha \in \Gamma$ then a is of type α for some $\alpha \in \Delta$. We say that Γ entails Δ in A, written $\Gamma \vdash_A \Delta$, if every token a of A satisfies $\langle \Gamma | \Delta \rangle$. If $\Gamma \vdash_A \Delta$ then the pair $\langle \Gamma | \Delta \rangle$ is called a constraint supported by the classification A."

According to the above definition of constraints, a sequent is a pair $\langle \Gamma | \Delta \rangle$ of sets of types from a classification. Following that, constraints provide regularities on type level in a classification. Together with the infomorphisms there are now mechanisms available that help us to align classifications from the different components in a distributed system with the help of an IF channel and based on regularities derived from some initial correspondences. The regularities within a distributed system are necessary to successfully establish a matching framework for this distributed system.

Channel theory: channels

The main aim in the application of the IF theory is the construction of a so-called IF channel. A channel is defined like follows (Barwise & Seligman, 1997; p.76):

"A channel C is an indexed family $\{f_i: A_i \rightleftarrows C\}_{i \in I}$ of infomorphisms with a common codomain C, called the core of C. The tokens of C are called connections; a connection c is said to connect the tokens $f_i(c)$ for $i \in I$."

This definition from Barwise and Seligman is a general definition for an n-ary channel with an index set {0,…,n-1}. However, most of the examples in the literature about the application of IF are dealing with two components only and this is exactly what we need for our application scenario. So we are talking about a binary channel and for the case of a binary channel, we can stick to the following channel definition that is given by (Schorlemmer & Kalfoglou, 2005):

"An IF channel consists of two IF classifications A_1 and A_2 connected through a core IF classification C via two infomorphisms f_1 and f_2."

A binary channel can be depicted like this:

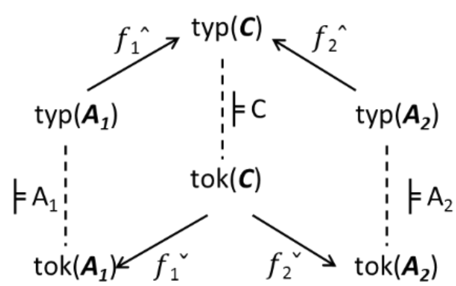

Fig. 3: Binary channel

According to this description, an IF channel consists of a core classification C, two component classifications A and B and corresponding infomorphisms that connect the core classification with the component classifications. Additionally, the core of the IF channel is a classification whose tokens are connections between the tokens from the component classifications and whose types are the disjoint union of the types from the component classifications that are involved in the IF channel.

Complex systems and environments

Cilliers introduced in (Cilliers, 1998) a description of complex systems based on a list of ten characteristics that are typical for a complex system. We will use these characteristics later on to justify why we think that the application area of smart manufacturing is a perfect example of a complex environment.

(Cilliers, 1998, p. 2) explains that it is hard to provide a working definition what complexity means, but that a general description of a complex system can be given by attempting to analyse the characteristics of these complex systems.

Table 1 shows the list of characteristics for a complex system taken from (Cilliers, 1998, pp. 3-5), summarises the explanation for these characteristics (ibid.), and gives an example where these criteria apply, which is also taken from (Cilliers, 1998, pp. 6-7). Cilliers (ibid.) frames the example of the economic system as a complex system with individual human beings—in their role as economic agents—as the elements of the complex system and draws the border of the system around a single country.

Table 1: Characteristics of a complex system

Characteristics of complex system and explanation	Example: the economic system
(i) Complex systems consist of a large number of elements.	(i) The economically active people in a country certainly comprise a large number of elements.
(ii) In order to constitute a complex system, the elements have to interact, and this interaction must be dynamic. A complex system changes with time. The interactions do not have to be *physical*; they can be thought of as the transference of *information*.	(ii) The various individuals interact by lending, borrowing, investing, and exchanging money for goods. These relationships change continually.
(iii) The interaction is fairly rich, i.e. any element in the system influences, and is influenced by, quite a few other ones. The behaviour of the system, however, is not determined by the exact amount of interactions associated with specific elements.	(iii) An economic agent interacts with a large number of the other elements: shops, banks, other agents.
(iv) The interactions themselves have a number of important characteristics. Firstly, the interactions are non-linear.	(iv) The interaction is non-linear: money can receive compounded interest; small investments can produce large returns (or vice versa).
(v) The interactions usually have a fairly short range, i.e. information is received primarily from immediate neighbours. Long-range interaction is not impossible,	(v) Economic agents primarily interact with others that are in their near vicinity (not necessarily in a spatial sense): local shops or providers of service, as well as their colleagues or partners.

Characteristics of complex system and explanation	Example: the economic system
but practical constraints usually force this consideration.	
(vi) There are loops in the interactions. The effects of any activity can feed back onto itself, sometimes directly, sometimes after a number of intervening stages.	(vi) The activity of an agent may eventually reflect back on itself. A good investment can produce good returns (positive feedback), and overspending can result in a shortage in the money supply (negative feedback).
(vii) Complex systems are usually open systems, i.e. they interact with their environment.	(vii) The economic system is certainly open. It is virtually impossible to draw its borders. It is continuously influenced by the political system, international relationships, the stability of the society, etc.
(viii) Complex systems operate under conditions far from equilibrium.	(viii) Since the economic system is driven by the dynamics of supply and demand it can never be in a state of equilibrium.
(ix) Complex systems have a history. Not only do they evolve over time, but their past is co-responsible for their present behaviour.	(ix) Economic systems are greatly influenced by their history. Today's prices largely depend on yesterday's.
(x) Each element in the system is ignorant of the behaviour of the system as a whole, it responds only to information that is available to it locally.	(x) An economic agent can only act on the available information. It does not know what all the other agents are doing.

We can derive from Cilliers' list of the characteristics of complex systems (Cilliers, 1998) that a complex system is not merely built of a huge number of elements, but that the complexity of a system arises from the vital interactions between the components of a system.

According to Cilliers (ibid.), some systems show a very large number of elements or components and perform a bunch of sophisticated tasks, e.g. a production line, but an expert is able to analyse them by dividing such a system into its constituent parts and examining the parts separately. A system that can be analysed with such a reductionist approach is a complicated system.

Cilliers (ibid.) argues that complex systems cannot be analysed by way of reductionism, because, as already said, a complex system is not only constituted by its components, but also by the relationships and interactions between these components and their elements. By breaking apart such a system in its constituent parts, the interactions and relationships between the components would also be cut, and thus the analytical method would destroy what it wants to understand.

We derive from Cilliers' description of complex systems (ibid.) that we have to have a look on the interactions and relationships between the components of a complex system if we want to understand the behaviour of such a complex system. Later on, we will describe how we are going to handle complexity based on the insights we inferred from Cilliers.

Smart manufacturing as a complex environment

In this section, we provide arguments why the application domain of manufacturing, especially the area of smart manufacturing, which we have introduced above, has to be considered as a complex environment. For this categorisation of smart manufacturing as a complex environment, we are using the characteristics introduced by Cilliers (1998) and shown in table 1 to describe a complex system.

Table 2 shows the characteristics of a complex system according to Cilliers (ibid.) as we know it from the previous section, but this time with the example of smart manufacturing mapped to these characteristics.

Table 2: Smart manufacturing categorised as a complex system

Characteristics of complex system and explanation	Example: smart manufacturing
(i) Complex systems consist of a large number of elements.	(i) The application area of smart manufacturing certainly consists of a large number of elements, e.g. countless different products that are produced with similarly countless machines and tools in a wealth of different production environments (Elmaraghy, Elmaraghy, Tomiyama, & Monostori, 2012).
(ii) In order to constitute a complex system, the elements have to interact, and this interaction must be dynamic. A complex system changes over time. The interactions do not have to be *physical*; they can be thought of as the transference of *information*.	(ii) The various companies involved in the manufacturing process are normally organised in supply chain networks with differing relationships of supplier and customer depending on the specific product that is produced. These relationships, as well as the members of the supply chain networks, change continually (Cheng, Chen, & Chen, 2014).
(iii) The interaction is fairly rich, i.e. any element in the system influences, and is influenced by, quite a few other ones.	(iii) A manufacturing company interacts with a large number of other companies and organisations in the supply chain: e.g. supplier of raw material, supplier of manufacturing equipment and tools, original equipment manufacturer (OEM), customers (Klibi, Martel, & Guitouni,

Characteristics of complex system and explanation	Example: smart manufacturing
	2010), banks, innovators, research organisations, et al.
(iv) The interactions themselves have a number of important characteristics. Firstly, the interactions are non-linear.	(iv) The interaction is non-linear what can be seen in a variety of concepts in manufacturing (Efthymiou, Pagoropoulos, Papakostas, Mourtzis, & Chryssolouris, 2012). For example, the bullwhip effect says that differing order behavior can yield to a variety of inefficiencies in the supply chain forcing suppliers to produce more than it is actually needed. Also, the sales forecasts are non-linear and have to react on globalised markets (Alony & Munoz, 2007).
(v) The interactions usually have a fairly short range, i.e. information is received primarily from immediate neighbours. Long-range interaction is not impossible, but practical constraints usually force this consideration.	(v) Manufacturing companies primarily interact with companies and organisations that are members of their own supply chain or ecosystem: e.g. suppliers or customers (Cheng et al., 2014).
(vi) There are loops in the interactions. The effects of any activity can feed back onto itself, sometimes directly, sometimes after a number of intervening stages.	(vi) The performance of a manufacturing company eventually reflects back on itself (Mason, Fowler, & Matthew Carlyle, 2002). Delivering products on time and in good quality that meets the requirements of the customer to a competitive price can lead to higher demand for the goods from the market. On the other side, a product with a too high sales price or in bad quality typically leads to decreasing demand from the market.
(vii) Complex systems are usually open systems, i.e. they interact with their environment.	(vii) The manufacturing area is certainly an open system. It is continuously influenced by a globalised market, new competitors, new technologies, demand for new products, et al. (Vogel & Lasch, 2016)
(viii) Complex systems operate under conditions far from equilibrium.	(viii) Since the manufacturing area is—similar to the economic system itself—driven by the dynamics of supply and demand (Cheng et al., 2014) it can never be in a state of equilibrium.

Characteristics of complex system and explanation	Example: smart manufacturing
(ix) Complex systems have a history. Not only do they evolve over time, but their past is co-responsible for their present behaviour.	(ix) Manufacturing companies and their supply chains are greatly influenced by the history of their interactions. For example, the rating of a supplier heavily depends on its performance in the past (Mason et al., 2002).
(x) Each element in the system is ignorant of the behaviour of the system as a whole, it responds only to information that is available to it locally.	(x) A manufacturing company can only act on the available information. It does not know what all the other manufacturing companies in the supply chain or even in other supply chains are doing.

This comparison of the application area of smart manufacturing with the characteristics of a complex system given by Cilliers (ibid.) is by far not comprehensive and complete; however, it shows that the domain of smart manufacturing meets all of Cilliers' (ibid.) criteria to be classified as a complex system or a complex environment.

METHODOLOGY

To find out relationships in complex environments, we chose the context of smart manufacturing as our application domain and we have seen in the previous section that this application area is a perfect example of a complex environment. The area of smart manufacturing is undergoing many research activities nowadays and one of the main ideas behind that concept is the idea that the production system shall reach a certain level of self-organisation and autonomy. We believe that for implementing and reaching this self-organisation and autonomy, it is necessary that we can predict the system's behaviour up to a certain degree. Therefore, we first must understand the overall system's behaviour in a specific context, for which we have to know about the relationships between the objects within the system under specific environmental conditions.

Thus, what we are aiming at, is to identify the *relationships of interest* in complex environments. By this term, we mean any kinds of relationships between objects in which people within the situation are interested and which these people can appreciate in terms of Vickers' notion of 'appreciative system' (Vickers, 1995) or which are among the 'affordance' in semiotics (Gibson, 1977) that the situation provides. We are inspired by the term of 'Situation of Interest' in the literature of Soft Systems Thinking, for example in the work of Checkland (Checkland, 2000) or Stowell (Stowell, 2016), and suggest the term 'Relationship of Interest' to show that our work even though highly technical has a philosophical origin of Interpretivism.

In the previous sections, we showed that various researchers already successfully applied IF in a series of different scenarios where the relationships of two or more sets of things have to be determined in environments that can be seen as distributed systems according to the definition shown in (Barwise & Seligman, 1997). All those application scenarios have in common that for the construction of the IF channel they follow the approach that is described in the diverse work of Kalfoglou and Schorlemmer, e.g. in (Schorlemmer & Kalfoglou, 2003b). In those examples, the construction of the IF channel normally starts at the baseline of a set of initial correspondences as a partial alignment between the IF classifications of the distributed system. Those initial correspondences either represent known relationships between the tokens of the involved classifications or relationships between the types of the classifications of a distributed system.

By applying Channel Theory to our application scenario according to these traditional approaches we observed some drawbacks (detailed shortly) caused by the higher level of complexity in this application scenario. These insights and shortcomings of the current application of IF are the motivation to examine how Channel Theory can be applied even in complex environments.

Defining and Handling Complexity

Following (Cilliers, 1998), we describe a complex system as an environment with a high degree of dynamics. These dynamics are caused by interactions between a huge series of objects within those environments. Additionally, the set of objects and their characteristics are not static but evolve over time; also, the relationships between the objects evolve and have an impact on how those objects interact with one another. Especially, as we are working with a distributed system, the sorts of interactions do not have to be physical; they also can result from transferring information within the overall system.

To understand and predict a system's behaviour, for a complicated system we can break the system apart into its components and analyse the structure and constraints of the single components. Putting all the components back together would then give us a complete understanding of the overall system, and we can predict the system's behaviour under specific circumstances even if the system consists of a huge number of single components.

As the components in complex environments show vital interactions between one another and the components and their characteristics, as well as the relationships between the components, are evolving over time, only a part of the overall system's behaviour can be analysed at a specific moment in time. Thus, the approach to understanding the system's behaviour, which we described for complicated systems, does not work for complex systems. However, we can achieve some understanding of the system's behaviour by studying the interactions between certain types of objects and their characteristics within the system and thereby getting insights on the relationships between those objects.

Knowing about the relationships between specific objects in complex environments gives us an understanding of how those objects behave and interact with one another under specific circumstances. We can use this information to reveal regularities that govern such kinds of behaviour and interaction between the objects and thus, we can predict the behaviour of parts of the overall system when particular surrounding conditions are met.

Thus, what we are aiming at is to identify relationships of interest in such kind of complex environments, which we describe as distributed systems. With the help of those relationships, we want to find out the regularities between the objects' interactions in a specific environment that help us to predict the system's behaviour and thus to tame the level of complexity in this system.

Identifying Relationships of Interest in Manufacturing by Using Channel Theory

We want to find relationships of interest in the application area of manufacturing. Manufacturing is an application area that shows many dynamics of the kind we described in the previous section as well as further characteristics of complex systems and thus can be perfectly seen as a complex environment. We have introduced our application scenario in the section "Related Research" and the situation, where the production system automatically assigns manufacturing equipment to the next production step is a perfect example of a complex environment in a distributed system. We will use this equipment assignment process as our application scenario to show how we are aiming to find the relationships of interest in complex environments by using the Channel Theory.

Iterative Usage of Channel Theory

As we said in the previous sections, to identify the relationships of interest in our application domain, we are applying the theory of Information Flow, also known as Channel Theory. This theory provides us with the mathematical tools that help us to describe the flow of information within a distributed system and thus to find out about the relationships and regularities within such kinds of systems. Kalfoglou and Schorlemmer developed the mathematical constructs of Information Flow into a systematic and mechanised methodology within an Information Flow based framework, which they call IF-Map (Kalfoglou & Schorlemmer, 2003). We use this approach and the work of successive researchers in the tradition of Kalfoglou and Schorlemmer as a starting point for our application of Information Flow—and thus for the identification of relationships of interest in the application area of smart manufacturing.

What we observe in the work of Kalfoglou and Schorlemmer and the researchers who build their approaches based on this work is, that they are mainly using examples with quite simple structures of the contexts with a fixed defined set of—often only a few—instances (referred to as *tokens* in the context of IF) and properties (referred to as *types* in the context of IF). Additionally in the examples of the literature, those types and tokens are not expected to change over time—showing us that those examples do not meet some of our main criteria for complex environments.

While applying the Channel Theory in the application domain of smart manufacturing, we realised that the so far known approaches from the literature that have been used in other application areas are not capable of addressing the higher level of complexity in this environment adequately. We observed that we should revise especially the usage of IF *classifications*, which are one of the main building blocks in constructing the *channel* to infer the relationships between objects in a distributed system. Furthermore, to cope with the high degree of complexity in our application domain, we introduced an iterative 2-step approach based on composite channels to derive the relationships of interest between the objects involved in the context of our example.

This 2-step approach results, on the one side, from the situation that our application example from manufacturing is not limited to only a few objects within the two contexts: product P and suitable machinery M, in order to produce the specific product P_x with the help of machinery M_y. The production of a specific product is typically split into a series of different production steps *PS* that must be conducted in sequence. The production steps themselves then have to be conducted by applying one specific production process *PP* with the help of the equipment that is intended for actually conducting a specific type of production process—see (DIN8580:2003-09, 2003) for a definition and a list of such kinds of production processes. Generally, in a production environment, there are different types of products that must be produced and therefore even more different types of production steps that have to be conducted. Additionally, there is a series of different production equipment and tooling *ET* that might be used to produce the products in a specific production environment. This environment results in a plethora of different production steps as well as a variety of different production equipment and tooling that must be taken into account for a specific manufacturing environment.

On top of this huge number of tokens that we have to consider within our contexts of interest in manufacturing, we have to respect that within those contexts different types might be needed to describe the various tokens within the same contexts. For example, in the context of equipment and tooling *ET*, the types are used to describe the specific production capabilities of specific production equipment in combination with specific tooling. In the case of a machining tool (the token), such production capabilities (the types) are for example travelling distances for machine axis, tooling system, or sizes of the machine table—which kinds of details of the equipment are really needed depends on the specific kind of equipment. When choosing the types, the various specifics and characteristics of all the equipment in a manufacturing environment have to be taken into account. This leads to a series of different types that are needed to classify all the various equipment and their characteristics decently. In the end, this results in a number of different types for different kinds of tokens in the same context.

Additionally, we have to cope with a dynamic environment within the manufacturing area as all those objects (tokens) within our two contexts production step *PS* and equipment and tooling *ET* are not fixed and are changing over time. Companies in the area of manufacturing are faced with changing products and increasing variants of those products, but also the tooling and equipment are updated or even exchanged over time. The results are changing contexts, where the tokens, as well as the types that are classifying the tokens, are altered over time—even worse, with the changing types and tokens also the relationships and the resulting regularities within our complex environments are changing over time as well.

Following the work that stands in the tradition of Kalfoglou and Schorlemmer, the types and tokens of a context are represented in IF classifications with the help of a table called ‘classification table’ like the one that is shown in Table 3.

Table 3: Typical Representation of IF Classification as a Simple Table.

	AG	PA	IND	FS	EUBD
r_1	1	1	0	0	0
r_2	1	0	1	0	0
r_3	1	0	0	0	0

r_4	0	0	0	1	1
r_5	0	0	0	1	0

Table 3 shows a typical example of the use of simple tables to represent IF classifications. In this example, which is taken from (Schorlemmer & Kalfoglou, 2003b), the five tokens 'r_1 to r_5' are responsibilities that are classified to ministry units (the types). In this example, these types are AG (Agencies), PA (Passport Agency), IND (Immigration and Nationality Directorate), FS (Foreign Secretary), and EUBD (European Union Bilateral Department). It can be seen that within this example, a well-defined set of ministry units is used to show the relationships between the responsibilities and the ministry units.

Unfortunately, things are not that easy in complex environments—the manufacturing example that we have described may show that. We assume that putting together in single IF classification tables all necessary information for a decent description of the contexts for production steps or production equipment in a specific manufacturing environment is not appropriate. Such an approach would result in big IF classification tables with a huge number of tokens and even a huge number of different types, which are classifying those tokens. Those big IF classifications may cause that the development of applications based on such kind of tables and the handling and maintenance of those tables as well might be too awkward and too error prone.

Thus, we realise that we should revise especially the usage of IF classifications when we want to identify relationships of interest by applying the Channel Theory in complex environments. Additionally, as IF classifications are one of the main building blocks in the application of Channel Theory, we also have to investigate how a novel way of using IF classifications may affect the construction of the channel itself. To address this, we introduce an iterative approach for constructing the channel to align two contexts in a distributed system. This iterative approach consists of two separate components, which are both addressing different aspects of complexity in our application environment. The first component is a 2-step approach for constructing the channel based on a series of IF classifications instead of one big IF classification and the second component is a learning system to maintain the knowledge about the regularities that govern the overall systems' behaviour.

Channel Theory based 2-step approach

With the Channel Theory based 2-step, we align production steps that are needed for the production of a specific product with the available equipment and tooling that is able to conduct the specific production step in a given production environment. Typical production environments show a variety of different production equipment and tooling and only some of them might be suitable for a specific production step—so, we split the selection process into two consecutive steps to reduce the complexity of selecting the proper equipment and tooling for a specific production step.

In the first step, we use a context production step *PS* (see Table 4) with a high-level description of the production step and a context production process *PP* (see Table 5) for a list of available production processes in the exemplary environment. In Table 4, a token $PS_{x\text{-}y}$ indicates production step y for product x.

Table 4: Context Production Step PS for Product P1 and P2.

Production Step PS	Trimming workpiece	Deburring workpiece	Making hole	Deburring hole	Finishing workpiece	Assembling	Quality control
PS_{1-1}	1	0	0	0	0	0	0
PS_{1-2}	0	1	0	0	0	0	0
PS_{1-3}	0	0	1	0	0	0	0
PS_{1-4}	0	0	1	0	0	0	0
PS_{1-5}	0	0	0	1	0	0	0
PS_{1-6}	0	0	0	1	0	0	0
PS_{2-1}	1	0	0	0	0	0	0
PS_{2-2}	0	1	0	0	0	0	0
PS_{2-3}	0	0	1	0	0	0	0
PS_{2-4}	0	0	0	1	0	0	0

Table 5: Context Production Process PP.

Production Process PP	Drilling	Gluing	Grinding	Milling	Nailing	Painting	Reaming	Riveting	Sanding	Sawing	Screwing	Sinking	Turning	Welding
PP_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PP_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
PP_3	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PP_4	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PP_5	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PP_6	0	0	0	0	0	1	0	0	0	0	0	0	0	0
PP_7	0	0	0	0	0	0	1	0	0	0	0	0	0	0
PP_8	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PP_9	0	0	0	0	0	0	0	0	1	0	0	0	0	0
PP_{10}	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PP_{11}	0	0	0	0	0	0	0	0	0	0	1	0	0	0
PP_{12}	0	0	0	0	0	0	0	0	0	0	0	1	0	0

PP_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	0
PP_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Within this first step, we are filtering out all production processes that might be used to conduct a specific production step. Based on so-called initial correspondences and with the help of the IF classifications, we are able to construct the IF channel according to the process that is well described for example in the work of Kalfoglou and Schorlemmer, e.g. in Kalfoglou & Schorlemmer (2003) or Schorlemmer & Kalfoglou (2003b) as well as in the literature from other researchers, who are standing in the tradition of Kalfoglou and Schorlemmers work, e.g. Xu & Feng (2012), Yang & Feng (2012), or Mantri (2013). Initial correspondences are the result of a priori knowledge or other kind of heuristics that tell us about alignments between types and tokens in our contexts that are already known. For our example, such initial correspondences might give us knowledge on which production processes PP_x were used in the past to conduct a specific production step from context *PS*.

In our example, we have the following initial correspondences on token-level:

PS_{1-1} <-> PP_{10}
PS_{1-2} <-> PP_{3}
PS_{1-3} <-> PP_{1}
PS_{1-3} <-> PP_{4}
PS_{1-4} <-> PP_{1}
PS_{1-4} <-> PP_{4}
PS_{1-5} <-> PP_{12}
PS_{1-6} <-> PP_{12}

Table 6 shows the core that resulted from the constructed IF channel *PS-PP* for a specific product P1, its production steps P_{1-x}, and the corresponding production processes PP_x. The table also indicates an example of the kind of production processes PP_x that might be used for a specific production step P_{1-x}.

Table 6: Constructed core for IF channel PS-PP.

	types from classification PS				types from classification PP				
PS-PP	Trimming workpiece	Deburring workpiece	Making hole	Deburring hole	Drilling	Grinding	Milling	Sawing	Sinking
<PS_{1-1};PP10>	1	0	0	0	0	0	0	1	0
<PS_{1-2};PP3>	0	1	0	0	0	1	0	0	0
<PS_{1-3};PP1>	0	0	1	0	1	0	0	0	0
<PS_{1-3};PP4>	0	0	1	0	0	0	1	0	0

<PS_{1-4};PP1>	0	0	1	0	1	0	0	0	0
<PS_{1-4};PP4>	0	0	1	0	0	0	1	0	0
<PS_{1-5};PP12>	0	0	0	1	0	0	0	0	1
<PS_{1-6};PP12>	0	0	0	1	0	0	0	0	1

In the second step, we align a more detailed description of the production steps PS_{1-x}—here with the example of the production steps PS_{1-3} and PS_{1-4} 'making hole'— (see Table 7) with the equipment and tooling that is related to the resulting production processes PP_1 'drilling' and PP_4 'milling' from step #1, see Table 8 and Table 9 respectively.

Table 7: Classification DD - Detailed Description of Production Steps "Making Hole".

$DD_{MakingHole}$	Requirements/Specifications, linear dimensions all in millimetre (mm)				
Production Step PS	Metal	Wood	Long hole	Hole diameter ≤ 13	Hole diameter > 13
PS_{1-3}	1	0	0	1	0
PS_{1-4}	1	0	0	0	1

Table 8: Classification $ET_{Drilling}$ for the Context Production Process Drilling.

$ET_{Drilling}$	Capabilities, linear dimensions all in millimetre (mm)			
Equipment / Tooling - ET	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13
ET_{D1-1}	1	0	1	0
ET_{D1-2}	1	0	0	1
ET_{D2-1}	1	0	1	0
ET_{D3-1}	1	0	1	0
ET_{D3-2}	1	0	0	1
ET_{D4-1}	0	1	1	0
ET_{D4-2}	1	0	1	0
ET_{D5-1}	1	0	1	0

Table 9: Classification $ET_{Milling}$ for the Context Production Process Milling.

$ET_{Milling}$	Capabilities, linear dimensions all in millimetre (mm)						
Equipment / Tooling - ET	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13	End mill cutter set	Tool holder ≤ 13	Tool holder > 13
ET_{M1-1}	1	0	1	0	0	0	0
ET_{M1-2}	1	0	0	1	0	0	0
ET_{M1-3}	0	1	1	0	0	0	0
ET_{M1-4}	0	1	0	1	0	0	0
ET_{M1-5}	0	0	0	0	1	1	0
ET_{M1-6}	0	0	0	0	1	0	1
ET_{M2-1}	1	0	1	0	0	0	0
ET_{M2-2}	0	0	0	0	1	1	0
ET_{M2-3}	0	0	0	0	1	0	1
ET_{M3-1}	0	0	0	0	1	1	0
ET_{M3-2}	0	0	0	0	1	0	1

Within this second step, we are now able to do the matching on more specific details as we are now dealing only with a subset of the available machines and tools based on the results of the preselection from step #1. We propose an approach where each of the detailed descriptions of the production steps and the machine categories is encapsulated in their own IF classifications, e.g. one separate classification for production step “making hole” and one classification for production step "deburring hole” or one separate classification for equipment and tooling “drilling” and one separate classification for equipment and tooling “milling”. This way, we can introduce specific types within the separate IF classifications and thus can better describe the specifics of the tokens within an IF classification without increasing the complexity within the IF classifications when using only one big IF classification per context.

Assuming the following initial correspondences, we can construct the channel $DD\text{-}ET_{Drilling/Milling}$ for the contexts $DD_{making\ hole}$, $ET_{Drilling}$, and $ET_{Milling}$ that is shown in Table 10.

Initial correspondences:

PS_{1-3} <-> ET_{D1-1}
PS_{1-3} <-> ET_{D4-2}
PS_{1-3} <-> ET_{M1-1}

PS_{1-4} <-> ET_{D1-2}
PS_{1-4} <-> ET_{M1-2}

Table 10: Constructed core DD-ET for the contexts $DD_{making\ hole}$, $ET_{Drilling}$, and $ET_{Milling}$.

	types from classification DD				types from classification $ET_{Drilling}$				types from classification $ET_{Milling}$						
$DD\text{-}ET_x$	Metal	Long hole	Hole diameter ≤ 13	Hole diameter > 13	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13	End mill cutter set	Tool holder ≤ 13	Tool holder > 13
<PS_{1-3};ET_{D1-1}>	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
<PS_{1-3};ET_{D4-2}>	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
<PS_{1-3};ET_{M1-1}>	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0
<PS_{1-4};ET_{D1-2}>	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0
<PS_{1-4};ET_{M1-2}>	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0

Learning system for updating regularities

A result from the construction of an IF channel as described in the 2-step approach in the previous section or as described in the work of Kalfoglou and Schorlemmer and other researchers is a list of constraints that capture part of the regularities that govern some situation that can be modelled as a distributed system. These regularities help us to understand the interactions between the objects within the system and help us to predict the system's behaviour. However, these regularities are based on our a priori knowledge about the relationships between some of the objects within our system, represented as initial correspondences that are the starting point for constructing the IF channel.

As long as our initial correspondences do not reflect all relationships of interest within our environment, we are talking about an alignment from a particular perspective, and thus our initial correspondences are only reflecting a limited solution space. Consequently, we are still faced with a certain degree of uncertainty and cannot predict the complete system's behaviour. In our example, this might mean, that we are getting only answers from the system for production steps and equipment that are based on a set of production steps and a set of equipment known already working together, that is to say, only those production steps and equipment that have a certain kind of relationships, such as one 'covers' another, with the former get looked at. Production steps and equipment that have different kinds of relationships with the former would not have been checked. In addition, new or so far unknown production steps and equipment that had not been used so far are not involved in the analysis and thus are not reflected within the alignment. So, what we are looking for is a set of initial correspondences that reflects as many relationships of interest as possible in order to tame the level of uncertainty and thus the complexity of our distributed system. We can increase the number of initial correspondences by updating them continuously when new knowledge becomes available, for example, new objects (tokens) come into the system, or existing tokens of the system are getting new characteristics (types).

Table 11 shows an example of the *core* of an IF channel from our application scenario constructed based on the following initial correspondences:

PS_{1-3} <-> ET_{D1-1}
PS_{1-3} <-> ET_{D4-2}
PS_{1-3} <-> ET_{M1-1}
PS_{1-4} <-> ET_{D1-2}
PS_{1-4} <-> ET_{M1-2}
PS_{2-3} <-> ET_{M1-5}

Table 11: Constructed Core DD-ET for the contexts $DD_{\text{making hole}}$, ET_{Drilling}, and ET_{Milling}.

types from classification DD					types from classification $ET_{Drilling}$				types from classification $ET_{Milling}$						
$DD\text{-}ET_x$	Metal	Long hole	Hole diameter ≤ 13	Hole diameter > 13	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13	HSS toolset	Wood drill set	Drill chuck ≤ 13	Drill chuck > 13	End mill cutter set	Tool holder ≤ 13	Tool holder > 13
<PS_{1-3};ET_{D1-1}>	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
<PS_{1-3};ET_{D4-2}>	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
<PS_{1-3};ET_{M1-1}>	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0
<PS_{1-4};ET_{D1-2}>	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0
<PS_{1-4};ET_{M1-2}>	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0
<PS_{2-3};ET_{M1-5}>	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0

We can see from this core that for the two products P_1 and P_2 several production steps PS_{1-x} and PS_{2-x} are listed and each of them is representing the production step "making hole". The core also shows the drilling or milling machine in combination with tooling that is able to conduct a specific production step. From this core we are able to derive for example the following constraints:

Metal, Hole diameter ≤ 13 $\vdash$ Drilling{HSS toolset, Drill chuck ≤ 13};
Metal, Hole diameter ≤ 13 $\vdash$ Milling{HSS toolset, Drill chuck ≤ 13};
Metal, Hole diameter > 13 $\vdash$ Drilling{HSS toolset, Drill chuck > 13};
Metal, Hole diameter > 13 $\vdash$ Milling{HSS toolset, Drill chuck > 13};

Metal, Hole diameter ≤ 13 ├ Drilling{HSS toolset,Drill chuck ≤ 13}, Milling{HSS toolset, Drill chuck ≤ 13};
Metal, Hole diameter > 13 ├ Drilling{HSS toolset, Drill chuck > 13}, Milling{HSS toolset, Drill chuck > 13};
Metal, Long hole, Hole diameter ≤ 13 ├ Milling{End mill cutter set, Tool holder ≤ 13};
Long hole ├ Milling;
Long hole, Drilling ├ ;
Metal ├ HSS tool set;
Long hole ├ End mill cutter set;
Hole diameter ≤ 13 ├ Drill chuck ≤ 13;
Hole diameter > 13 ├ Drill chuck > 13

We are now able to use these constraints in a new iteration of our 2-step based construction of the channel for a so far unknown product P_3. Product P_3 has amongst others the production steps PS_{3-3}, PS_{3-4}, and PS_{3-5}, which are all standing for making holes, see Table 12 for a list of all production steps of products P_1, P_2, and P_3 that are concerned with making hole.

Table 12: Classification $DD_{Making\ Hole}$ for Products P_1, P_2, and P_3.

$DD_{MakingHole}$	Requirements/Specifications, linear dimensions all in millimetre (mm)				
Production Step PS	Metal	Wood	Long hole	Hole diameter ≤ 13	Hole diameter > 13
PS_{1-3}	1	0	0	1	0
PS_{1-4}	1	0	0	0	1
PS_{2-3}	1	0	1	1	0
PS_{3-3}	1	0	0	1	0
PS_{3-4}	1	0	0	0	1
PS_{3-5}	1	0	1	1	0

From the constraints of the core shown in Table 11, we have learned that for making holes for product P_1 and product P_2 that is to drill a hole in a metal workpiece with a diameter ≤ 13mm (PS_{3-3}), we need either a drilling machine with an HSS tool set and a drill chuck ≤ 13mm or a milling machine with an HSS tool set and a drill chuck ≤ 13mm. Furthermore, we have learned from the constraints that to make a hole in a metal workpiece with a diameter > 13mm (PS_{3-4}), we need either a drilling machine with an HSS tool set and a drill chuck > 13mm or a milling machine with an HSS tool set and a drill chuck > 13mm. To produce a long hole in a metal workpiece with a diameter ≤ 13mm (PS_{3-5}), we derive from the constraints that we need a milling machine with an end mill cutter set and a tool holder ≤ 13mm.

These constraints bring us to the following initial correspondences for product P_3 and its production steps PS_{3-3}, PS_{3-4}, and PS_{3-5} based on the a priori knowledge from the successful production of product P_1 and P_2:

PS_{3-3} <-> ET_{D1-1}
PS_{3-3} <-> ET_{D2-1}
PS_{3-3} <-> ET_{D3-1}
PS_{3-3} <-> ET_{D4-2}
PS_{3-3} <-> ET_{D5-1}
PS_{3-3} <-> ET_{M1-1}
PS_{3-3} <-> ET_{M2-1}
PS_{3-4} <-> ET_{D1-2}
PS_{3-4} <-> ET_{D3-2}
PS_{3-4} <-> ET_{M1-2}
PS_{3-5} <-> ET_{M1-5}
PS_{3-5} <-> ET_{M2-2}
PS_{3-5} <-> ET_{M3-1}

What we can see from the list of these initial correspondences, which does not even reflect the initial correspondences we already had for the production steps of product P_1 and P_2, is that we have now far more initial correspondences as a starting. This updated list is a result from our translation of the constraints from the core in Table 11 to new initial correspondences for the production steps of product P_3 while we were iterating again through the process of constructing the IF channel. These new and updated initial correspondences now take into account equipment and tooling that might not been used for product P_1 and P_2 so far but might be able to do the necessary job. We found these new initial correspondences on token-level, which we also call pairings, by applying the known constraints, which are in fact relationships on type-level, to those tokens, which are classified by the same types and thus belong to the same constraints—these kinds of tokens are known in Channel Theory as indistinguishable tokens (Barwise and Seligman, 1997, p. 71). This way, we have enlarged the number of our initial correspondences and thus the solution space for the relationships of interest within our system. This increased number of known relationships results, at the end, in a reduction of the complexity of the distributed system we are dealing with.

RESULTS

Complex environments show a high degree of dynamics caused by vital interactions between the objects within those environments and the alterations the set of objects and their characteristics within those environments go through over time. Manufacturing is an area where those kinds of dynamics are quite obvious, for example, manufacturing companies increasingly have to manufacture new products and variants of products or to integrate new equipment or machinery into an existing production system.

We show that we can tame the level of complexity in dynamic environments by identifying relationships of interest between the objects in such environments. Knowing about the relationships that are relevant to a particular task or of a particular interest between the objects in complex environments gives us insights on how those objects behave and interact with one another under specific circumstances. We can use this information to reveal regularities that govern such kind of behaviour and interaction between the objects and thus can predict the behaviour of the overall system when particular surrounding conditions are met.

To identify a type of relationship of interest between the objects in a specific complex environment, we apply the theory of Information Flow (IF for short), also known as Channel Theory put forward by Barwise and Seligman (1997). We chose the Channel Theory as it is a solid theory, well designed for modelling things that form a wholeness as distributed systems, and our application area of complex environments can perfectly be seen as such a distributed system. Furthermore, there exists a series of applications based on the Channel Theory that proves that this theory is able to find relationships between two or more sets of objects from different contexts in a distributed environment.

While applying the Channel Theory in the application domain of manufacturing, we realise that the so far known approaches from the literature that have been used in other application areas are not capable of addressing the higher level of complexity in this environment adequately. To cope with the high degree of complexity in our application domain, we introduce an iterative 2-step approach based on composite channels to derive the relationships of interest between the production steps that have to be conducted and the production capabilities of the available equipment. Furthermore, we apply the Channel Theory iteratively for every new or so far unknown production step. Within this iterative use of Channel Theory, we convert the regularities that we have learned from constructing the IF channel into new so-called initial correspondences that reflect our current a priori knowledge about the relationships between the objects within our distributed system. This way, we enhance the list of initial correspondences. The more we know about those relationships, the better we can predict the system's behaviour.

By enhancing the way how the Channel Theory has been applied so far with our 2-step and iterative approach we show with the help of an example from the manufacturing domain that the Channel Theory can also be applied successfully in complex environments to identify relationships of interest.

With the above example and discussion, we are now in the position to further clarify what we mean by the term of 'relationship of interest'. First of all, the so-called initial correspondences can be anything in which we are interested. For example, we might be interested in that two objects are similar, or completely different, or one relies on the other. That is, a correspondence can be any association between two objects in which we are interested. Furthermore, how other objects are associated with such initial correspondences can also be any kind in which we are interested. That is, how a third object is associated in any particular kind with a pair of objects that one thinks have a given association. For example, a school A offers a course C1 that is deemed a prerequisite for another course C2 offered by another school B. Through an analysis like the one described in this paper, we would be able to identify that the two schools A and B have a relationship that each offers at least one course that takes at least one course offered by the other as a prerequisite or vice versa.

REFERENCES

Alony, I., & Munoz, A. (2007). The bullwhip effect in complex supply chains. In *International Symposium on Communications and Information Technologies ISCIT 2007* (pp. 1355–1360). Retrieved from http://ieeexplore.ieee.org/document/7123235/

Barwise, J., & Seligman, J. (1997). *Information Flow: The Logic of Distributed Systems* (reprinted). New York: Cambridge University Press.

Checkland, P. (2000). Soft Systems Methodology : A Thirty Year Retrospective. *Systems Research and Behaviural Science*, *58*(17), 11–58. https://doi.org/10.1002/1099-1743

Cheng, C. Y., Chen, T. L., & Chen, Y. Y. (2014). An analysis of the structural complexity of supply chain networks. *Applied Mathematical Modelling*, *38*(9–10), 2328–2344. https://doi.org/10.1016/j.apm.2013.10.016

Cilliers, P. (1998). *Complexity & Postmodernism: Understanding Complex Systems*. London: Routledge.

DIN8580:2003-09. (2003). Norm DIN 8580:2003-09, Fertigungsverfahren - Begriffe, Einteilung (engl. manufacturing processes - terms and definitions, divisions). Berlin: Beuth Verlag GmbH.

Efthymiou, K., Pagoropoulos, A., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2012). Manufacturing systems complexity review: Challenges and outlook. *Procedia CIRP*, *3*(1), 644–649. https://doi.org/10.1016/j.procir.2012.07.110

Elmaraghy, W., Elmaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology*, *61*(2), 793–814. https://doi.org/10.1016/j.cirp.2012.05.001

Feng, J., & Xu, K. (2012). An information flow based approach to finding informational relationships between questionnaires. In *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)* (pp. 1–4). Institution of Engineering and Technology. https://doi.org/10.1049/cp.2012.2261

Kalfoglou, Y., & Schorlemmer, M. (2003). IF-Map : An Ontology-Mapping Method based on Information-Flow Theory. *Journal on Data Semantics*, *1*(1), 98–127.

Kalfoglou, Y., & Schorlemmer, M. (2005). Using Formal Concept Analysis and Information Flow for Modelling and Sharing Common Semantics: Lessons Learnt and Emergent Issues. In F. Dau, M.-L. Mugnier, & G. Stumme (Eds.), *ICCS 2005* (1st ed., Vol. LNAI 3596, pp. 107–118). Springer-Verlag Berlin Heidelberg.

Kalfoglou, Y., & Schorlemmer, M. (2010). The Information Flow Approach to Ontology-Based Semantic Alignment. In F. Dau, M.-L. Mugnier, & G. Stumme (Eds.), *Theory and Applications of Ontology: Computer Applications* (1st ed., pp. 101–114). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-90-481-8847-5_4

Klibi, W., Martel, A., & Guitouni, A. (2010). The design of robust value-creating supply chain networks: A critical review. *European Journal of Operational Research*, *203*(2), 283–293.

https://doi.org/10.1016/j.ejor.2009.06.011

Koren, Y. (2010). *The Global Manufacturing Revolution – Product-Process-Business Integration and Reconfigurable Systems*. (A. Sage, Ed.). Hoboken, New Jersey: John Wiley & Sons, Ltd.

Li, W., & Mehnen, J. (Eds.). (2013). *Cloud Manufacturing*. London: Springer London. https://doi.org/10.1007/978-1-4471-4935-4

Mantri, R. (2013). *Database Web Service Composition based on the Notion of ' Informational Relationship .'* PhD Thesis. University of the West of Scotland, Paisley.

Mason, S. J., Fowler, J. W., & Matthew Carlyle, W. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, *5*(3), 247–262. https://doi.org/10.1002/jos.102

Mellal, N., & Dapoiny, R. (2007). Formal Method for Automatic and Semantic Mapping of Distributed Service-Ontologies. In *Proceedings of the Second International Conference on Software and Data Technologies* (Vol. Volume 1:, pp. 259–263). SciTePress - Science and and Technology Publications. https://doi.org/10.5220/0001329602590263

Papakostas, N., Efthymiou, K., Georgoulias, K., & Chryssolouris, G. (2012). On the configuration and planning of dynamic manufacturing networks. *Logistics Research*, *5*(3–4), 105–111. https://doi.org/10.1007/s12159-012-0086-9

Schorlemmer, M., & Kalfoglou, Y. (2003a). On Semantic Interoperability and the Flow of Information. In *ISWC'03 Workshop on Semantic Integration* (p. 86). CEUR Workshop Proceedings.

Schorlemmer, M., & Kalfoglou, Y. (2003b). *Using Information-Flow Theory to Enable Semantic Interoperability*. (I. Aguiló, L. Valverde, & M. T. Escrig, Eds.), *Artificial Intelligence Research and Development*. Amsterdam: IOS Press.

Schorlemmer, M., & Kalfoglou, Y. (2005). Progressive ontology alignment for meaning coordination: An Information-Theoretic Foundation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05* (pp. 737–744). New York, New York, USA: ACM Press. https://doi.org/10.1145/1082473.1082586

Schorlemmer, M., & Kalfoglou, Y. (2008). Institutionalising ontology-based semantic integration. *Applied Ontology*, *3*(3), 131–150. https://doi.org/10.3233/AO-2008-0041

Stock, D., & Bildstein, A. (2015). Cloud-Based Interoperability for Dynamic Manufacturing Networks. In *Enterprise Interoperability* (pp. 122–127). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119081418.ch17

Stowell, F. (2016). Soft, Not Vague. In Baecker (Ed.), *Schlüsselwerke der Systemtheorie* (pp. 373–400). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-531-20004-0_30

Vickers, G. (1995). *The Art of Judgment: A Study of Policy Making* (Centenary). SAGE Publications, Inc.

Vogel, W., & Lasch, R. (2016). Complexity drivers in manufacturing companies: a literature review. *Logistics Research*, *9*(1), 1–66. https://doi.org/10.1007/s12159-016-0152-9

Wang, Y., & Feng, J. (2007). FCA assisted IF channel construction towards formulating conceptual data modeling. *Wseas Transactions On Systems*, *6*(6), 1159–1167.

Yang, G. (2015). *Achieving Database Semantic Interoperability with a Perspective of Information Flow and Formal Concepts*. PhD Thesis. University of the West of Scotland.

Yang, G., & Feng, J. (2012). Database Semantic Interoperability based on Information Flow Theory and Formal Concept Analysis. *International Journal of Information Technology and Computer Science*, *4*(7), 33–42. https://doi.org/10.5815/ijitcs.2012.07.05