Institute of Architecture of Application Systems

Costume Languages as Pattern Languages

Johanna Barzen, Frank Leymann

Institute of Architecture of Application Systems,
University of Stuttgart, Germany

{barzen, leymann}@iaas.uni-stuttgart.de

BibTeX

```
@inproceedings {INPROC-2015-38,
    author      = {Johanna Barzen and Frank Leymann},
    title       = {{Costume Languages as Pattern Languages}},
    booktitle   = {Proceedings of PURPLSOC (Pursuit of Pattern Languages for
                  Societal Change). The Workshop 2014},
    editor      = {Peter Baumgartner and Richard Sickinger},
    address     = {Krems},
    publisher   = {PURPLSOC 2015},
    pages       = {88-117},
    type        = {Workshop-Beitrag},
    month       = {Juni},
    year        = {2015},
    isbn        = {978-3-7375-5458-9},
    language    = {Englisch},
    url         = {http://www2.informatik.uni-stuttgart.de/cgi-
bin/NCSTRL/NCSTRL_view.pl?id=INPROC-2015-38&engl=}
 }
```


Universität Stuttgart
Germany

Costume Languages as Pattern Languages

Barzen, Johanna, Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, Barzen@iaas.uni-stuttgart.de

Leymann, Frank, Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, Leymann@iaas.uni-stuttgart.de

Abstract:

In media science, the concept of the costume language is only vaguely defined. We propose a formal definition of this concept, which is based on the concept of pattern languages. In order to derive the patterns of such a language we introduce a method that supports capturing knowledge about clothes, identifying costumes based on clothes and abstracting costumes into costume patterns. Thus, costumes are understood as concrete solutions to the recurring problem of achieving effects in films by using clothes. Associations between patterns and concrete solutions are maintained to ease the development of concrete solutions from a pattern's abstract solution description. Our method and its associated formalizations are generalized to support the identification of patterns from concrete solutions in other domains.

Keywords: *Costumes; Costume Languages; Vestimentary Communication; Formal Languages; Digital Humanities.*

ISSN (tba)
www.purplsoc.org

1. Introduction

A close examination of the natural sciences and engineering shows that the use of concepts, methods and technologies of computer science is in an advanced stage (Hey, Tansley, & Tolle, 2009). This relation between these sciences is reflected by the term “eScience”. In comparison, the use of techniques and methods of computer science in the humanities (called eHumanities or digital humanities (Berry, 2012; Burdik, Drucker, Lunenfeld, Presner, & Schnapp, 2012; Terras, Nyhan, & Vanhoutte, 2013)) is rather rudimentary. While databases, archives and document systems as well as the technologies of computational linguistics and visualization are used in some areas of the humanities, the advanced use computer science-derived concepts, methods and technologies are still marginal. In this contribution, we present a method (Section 2) to derive costume languages in movies based on the concepts of ontologies and pattern languages, which are both quite frequently used in computer science. Furthermore, we propose to concretize the term “costume language” by using the concept of “formal languages” from computer science (Section 3). This formalization supports the method introduced before and clarifies the rather vague term “costume language” in media science. The implementation of a tool environment to support our method is described in an accompanying article (Fehling, Barzen, Falkenthal, & Leymann, 2014). In section 4 we generalize our formalization and argue that it is applicable in many other domains. Altogether, we show some of the advantages of using formal languages and pattern languages in the Digital Humanities.

Thus, this paper combines various different aspects of costume languages reaching from methodological aspects to formal aspects. The methodological aspects (section 1 and 2) are targeted to readers from humanities, while the formal aspects (sections 3 and 4) assume some background in mathematics.

1.1. The Problem

For the media science detecting a systematic representation of costumes in films is of great interest. The costume, as the enclosing textile shell of the actor’s body, is a strong visible sign and an important tool for designing the diegetic world of a movie (Hollander, 1993). It adapts our “real world” signs of clothes and provides important information about the characters, their profession, social standing, moods, and character traits. To communicate this information the costume uses “vestimentary communication”.

The term “vestimentary” has its etymological origin in the Latin word for clothing: “vestimentum”. It refers to the communication that takes place through clothes. Especially in movies the costume is deliberately used as a multi-layered instrument of manipulation and contains important information about the characters, their transformations and developments, as well as the time and setting of the movie (Bruzzi, 1997; Cook, 1996; McDonald, 2010). This cinematic vestimentary communication is referred to as “costume language” in the literature (Devoucoux, 2007; Giannone, 2005).

Even if the term “costume language” is used quite frequently, it is usually meant in a metaphorical sense and most of the time is based on evidence from several individual film analyses. Even if the communicative capabilities of the costume are beyond dispute, the detection of a specific costume language is known to be difficult (Burger, 2002; Giannone, 2005).

However, when focusing on the costumes of western genre movies for example, we can identify stereotypically used clothes that allow the recipient to distinguish villains from heroes. The ability to make this distinction is based on the fact that recipients recognize stereotypes and the convention to use quite similar costumes for the same stereotypes (Giannone, 2005). However, in practice, neither have the stereotypical costumes been described in detail, nor have they been used and analyzed to derive a concrete costume language. Especially when looking at the minor roles and the extras that are constructed for a quick understanding (next to the main characters that aim for authentic individualization) it seems to be promising to abstract this cinematic vestimentary communication of western genre movies into a pattern language of costumes: we define costume patterns as proven solutions to recurring problems of a costume designer to communicate a specific character, such as finding the adequate textile expression for the role of a certain cowboy or saloon lady for example.

1.2. Related Work

A first systematic and (in a certain sense) formal approach to the investigation of costumes was articulated by Burger (2002). Burger tried to apply Barthes' (1985) semiotic approach to fashion to film costumes. However, Burger's proposed system of pre-defined templates for describing costumes from is closer to an inventory system than a method for analyzing given costumes and their relations in detail. Especially, these templates are not suitable for generating new insight in understanding the tasks and functions of costume languages.

A rather different approach, developed by Lurie (2000), created an analogy between a language of clothes and natural language. In her approach the individual pieces of clothes, such as a shirt or a pair of trousers provided the "vocabulary" of the language of clothes, while a set of clothes such as a special outfit was conceptualized as a "statement". In contrast to our work, Lurie (2000) focused only on clothes and not on costumes and compared the language of clothes to natural language instead of formal language. Unlike natural language the language of clothes is an indirect and non-verbal communication and Lurie (2000, p. 19) notes that many elements of natural language cannot be transferred to a language of clothes. Therefore in this article we use the model of the formal rather than the natural language to develop a language of clothes.

In using the concept of formal languages, we refer to established and proven concepts from computer science. Going beyond Lurie (2000) we propose granules finer than complete pieces of clothes that we call *primitives* (Section 2); primitives are used for a comprehensive description of clothes: For example, a black shirt with a standing collar has a completely different communicative effect than a pink and white striped shirt with a Peter Pan collar. While both are "just" shirts and therefore a more precise description is needed to make them communicatively precise. A piece of clothes such as a shirt is referred to as *base element* in our approach.

Barzen, Leymann, Schumm, & Wieland (2012) and Schumm, Barzen, Leymann, & Ellrich (2012) illustrate how costume patterns can be used to abstract cinematic vestimentary communication and to capture the "essence of convention" of how costumes are used in films. In Barzen et al. (2012) there is a first pattern-based approach towards IT support of costume management in film, as well as a first attempt at creating a costume pattern format. This approach was extended in Schumm, Barzen, Leymann, & Ellrich (2012), who describe

the composition of costumes from their base elements[1] as the core of each costume pattern and provides the base for the proposed method and formalization introduced in this article. We describe the composition of clothes by formal languages generated by grammars and costume patterns as equivalence classes. By treating patterns as classes, the connection between patterns and the concrete items they are abstracted from are maintained.

In Taibi (2006) pattern of the design of distributed object systems are described by formulas of the language of first order predicate logic. The behavior of such patterns is described in this approach by a temporal logic of actions. Therefore, combinations of patterns can be verified with respect to accuracy and correctness. Taibi (2006) reviews further formalizations of patterns proposed by other authors: All these approaches have in common that it is assumed that the patterns to be assembled are known and that the combinations of patterns are to be checked for correctness. In our approach we focus on the identification of appropriate patterns as well as finding known solutions that either fit or that can serve as a template to develop new solutions; verification of composite patterns is not relevant in our domain. For this reason, the approaches that are reviewed in Taibi (2006) in addition to the approach proposed in Taibi (2006) itself is not suitable for solving our problem.

The use of grammars to describe valid selection sequences in a pattern language is proposed in Zdun (2007). The relations between the patterns of the pattern language are annotated with discrete values. Such a discrete value represents the influence that the selection of the next pattern has to the expected quality of the overall solution. Therefore, the quality of the solution produced by a sequence of several patterns can be evaluated. Furthermore, the relations are enriched with information about whether the next pattern can be applied (optional), or must be applied (mandatory), or if the next pattern is a variant of the previous pattern. The descried diagram of the pattern language is then transformed into a grammar, while the production rules arise essentially from the relations between the patterns. The temporal sequences using patterns that achieve a solution with a certain quality are the words of this language. As mentioned above, for our approach the identification of individual patterns is essential. Composite patterns (i.e. compositions of costumes) are not relevant in our domain. In our approach we use grammars to aggregate the individual concrete solutions resulting in a pattern itself. Such a concrete solution is some sort of instance of the abstract solution represented by the corresponding pattern. Specifying relations between patterns as temporal sequences as used in Zdun (2007) is insufficient: In our domain we require the ability to specify relations of any kind between patterns, which is why we support the definition of sets of base relations with different semantics.

The kind of information to document more details about the solutions offered by patterns in the domain of software architecture has been described in Zdun & Avgeriou (2008): A UML profile has been defined for this purpose. In our approach, we are based on the document structure introduced by Alexander, Ishikawa, & Silverstein (1977) to present pattern languages and the solution provided by patterns; this document structure has been used in many other pattern domains with a few specific modifications or extensions, respectively. In contrast to Zdun & Avgeriou (2008) we are not specifying the abstract solution of a pattern in a structured manner but the corresponding concrete solutions. These concrete solutions have been captured to support the detection and abstraction of patterns or they are built by following the description provided by the abstract solution of a pattern, respectively.

[1] In our previous work (Schumm, Barzen, Leymann, & Ellrich, 2012) we referred to the base elements as primitives but here we have refined our terminology.

2. Method for Deriving Costume Languages

As mentioned in the introduction, describing a costume language is a rather difficult task. As a starting hypothesis we assume that each genre has a specific costume language with established conventions. These established conventions could be, for example, the similarity of costumes of frequently recurring stereotypes or the use of the same colors or certain materials for certain character traits. The assumption that a costume language proves to be dependent on the genre is due to the often similar set of characters and the resembling of style, used colors and aesthetics, which can be found in films of the same genre. It may even be that a costume language of a certain film of a genre can deviate from the language of another film of the same genre. But by containing similar elements they can be combined into a costume language of the genre. Therefore, to extract these similarities and differences, we formalize both, the concepts of a language of clothes as well as the concepts of a costume language.

Below, we introduce our method for determining the relevant components for a costume language. To illustrate the relation of clothes and costumes, section 2.1 addresses the process by which a costume designer creates a costume. Section 2.2 concentrates on the systematic way to formally describe clothes, while section 2.3 illustrates the derivation of concrete costumes and their abstraction into costume patterns. Section 2.4 describes a consolidated method that combines the capturing of knowledge about costumes and their abstraction into costume patterns.

2.1. Designing Costumes: The Act of Achieving Effects of Clothes

To get a better understanding of how a costume can be defined and what differences exist between clothes and costumes, we want to take a closer look on how a costume is created. Figure 1 briefly sketches the creative act of a costume designer in determining the right costume for a character. At the beginning of the process of designing a costume, the costume designer discusses the concepts and aesthetics of the film with the director, art director and other members of the production team and reads the script of the film to extract the characters and information about these characters. Based on this information the costume designer selects clothes that make a certain statement about the character, character traits, or the setting of the film, for example. This selection of some special clothes to communicate and support content gives the clothes an intended effect. Clothes and its intended effect together are defined as a costume.

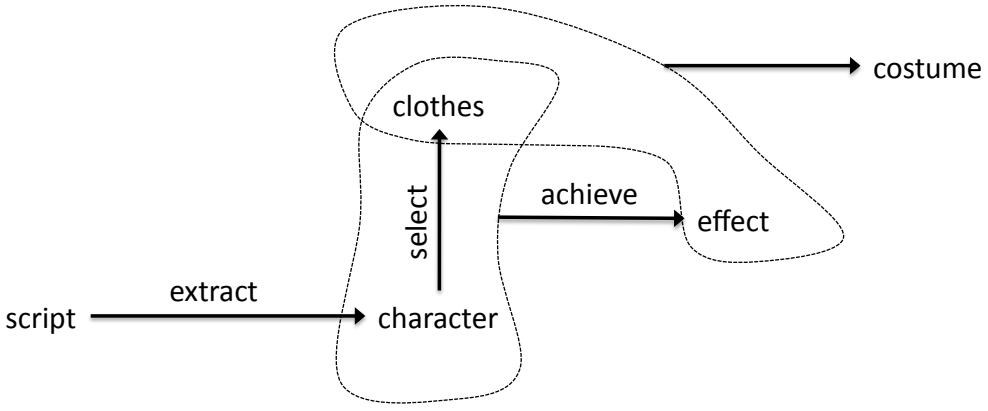

Figure 1: A Costume as a set of clothes with intended effect

However, whether the clothes really provoke the intended effect when viewed by the audience is not important. The effect at this stage is the effect that the costume designer assumes due to experience and expertise, as well as individual taste and dependents on the situation (the director, budget or setting, for example). Whether or not this intended effect matches the effect of the costume language is therefore not relevant since the costume language aims to reflect established conventions used to communicate by costumes. In Section 3.3 we will take a closer look at the effect of costumes.

2.2. Modeling of Clothes: Ontologies, Primitives and Base Elements

Every costume consists of its haptic basis, namely "clothes". Thus, we first introduce a systematic way to capture clothes in an ontology (see Figure 2). The term *clothes* is used in the sense of a complete outfit. An example is the total outfit of a sheriff. It consists of specific items like a cowboy hat, boots, a vest, a belt, a sheriff's star and so on. We refer to these items as *base elements*. Each base element is in turn composed of *primitives*. Trousers, for example, have two legs, a waistband and a closure, which in turn can be made from a zipper, buttons, or both. The difference between a base element and a primitive is that the base element, although composed of primitives, provides a whole separate part of a costume while the primitives typically need other primitives to aggregate into a base element. The distinction between base elements and primitives is of importance because the granularity of the base elements is not sufficient to describe all the needed information. A "collar" as a primitive of a base element "shirt" for example, may have different colors or materials than the rest of the base element. This fact has an influence on its communicative effect. In Section 3, the terms "base element" and "primitive" are formalized in terms of an alphabet of a language of clothes and a language of primitives.

Note that the concept of subpatterns is not appropriate in the domain of costume languages: The composition of a costume out of concrete base elements and primitives is important for a costume language. When specifying base elements as subpatterns, such base elements would become generic because patterns are abstract, i.e. templates for producing concrete entities instead of being the concrete entities themselves.

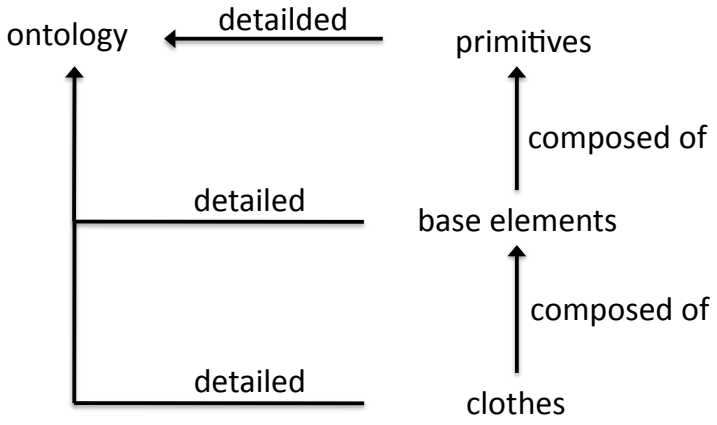

Figure 2: Method for the detection and description of clothing

The composition of clothes out of base elements and primitives supports a differentiated and precise description of clothes. A set of base elements and primitives can thus be combined in numerous ways, resulting in many different clothes. Every textual description of clothes reduces visual and haptic information. Furthermore, there are often many different ways to describe a certain piece of clothes. Therefore, the use of an ontology facilitates the

consistent naming of base elements and primitives during the analysis of films by predetermining names. In general, the concept of *ontology* supports creating a unified vocabulary of a domain and supports to structure knowledge in a systematic way to make this knowledge available and reusable by others. Through the use of established ontology languages such as RDF (Resource Description Framework), RDFS (Resource Description Framework Schema) and OWL (Web Ontology Language), this semantically linked knowledge in turn provides the opportunity to use query languages such as SPARQL to analyzed the captured knowledge.

In order to approach such a comprehensive ontology of the domain of clothes we first structured the relevant individual areas of clothes as *taxonomies* (Barzen, 2013). These taxonomies provide the systematization of (i) the individual pieces of clothes, i.e. the base elements, (ii) possible characteristics of the clothes such as their composition from primitives, their material, their color, their design, their state, and their shape, and (iii) the potential relations between their base elements such as their dressing order. The classes defined in these taxonomies (such as hat, collar etc.) and their sub-classes (such as wool cap, Kent collar) serve as concepts of the ontology and are refined by adding attributes to these concepts and as well es relations between the classes. For example, the concept “shoe” can be related with the concept “sock” by the relation “worn above”. This ontology supports to describe each piece of clothes in all its peculiarities in a fast, flexible and accurate way (Barzen, 2013).

The most important taxonomies are those of base elements and primitives. They represent the core of the ontology as well as the basis of a language of clothes. Their concepts have attributes like color, shape, design or material. The ranges of values of these attributes are also represented by taxonomies (Barzen, 2013). For example, a base element shirt, consist of the primitives white, short stand-up collar with blue stripes, a light blue front and back and a white button border with dark blue buttons. All those attributes such as the specific form, the design and material, the current state and the way a piece of clothes is worn potentially influence the effect of a costume. Therefore they need to be described in detail as facilitated by our ontology.

2.3. Deriving Costume Patterns

Figure 3 illustrates how to derive costume patterns. First, in order to yield a relevant quantity of costumes, a film corpus is chosen. For this purpose, several movies of a film genre are selected which reflect a representative cross section of this genre. To get this representative cross section well-defined criteria have to be specified to identify these films.

The recurrence of similar costumes indicates that an abstraction of these costumes into costume patterns is possible. To determine the recurrence of costumes, recurring characters need to be identified, and their clothes, stereotype and context must be described. Because a costume consists of clothes with intended effect in order to transport information by clothes (see section 2.1), in the next step the costumes are identified. To enable later data analysis all this information is captured as instances in the ontology.

A solution documented by a pattern is abstract in the sense that it is free of context to be applicable in many different situations. In contrast to this, a captured costume is a concrete solution for a specific design problem the costume designer had to face when creating the costume. It is “concrete” in the sense that it specifies details like color, material etc. used. However, a costume pattern documents an abstract solution. Thus, it consolidates all

concrete solutions and represents their “essence”. Therefore, the concrete costumes need to be grouped into costumes with the same effect. To group costumes, the attributes of the costumes as well as the stereotype they represent and their context such as the story or the character traits needs to be considered. Costumes with the same effect can then be represented as a costume pattern.

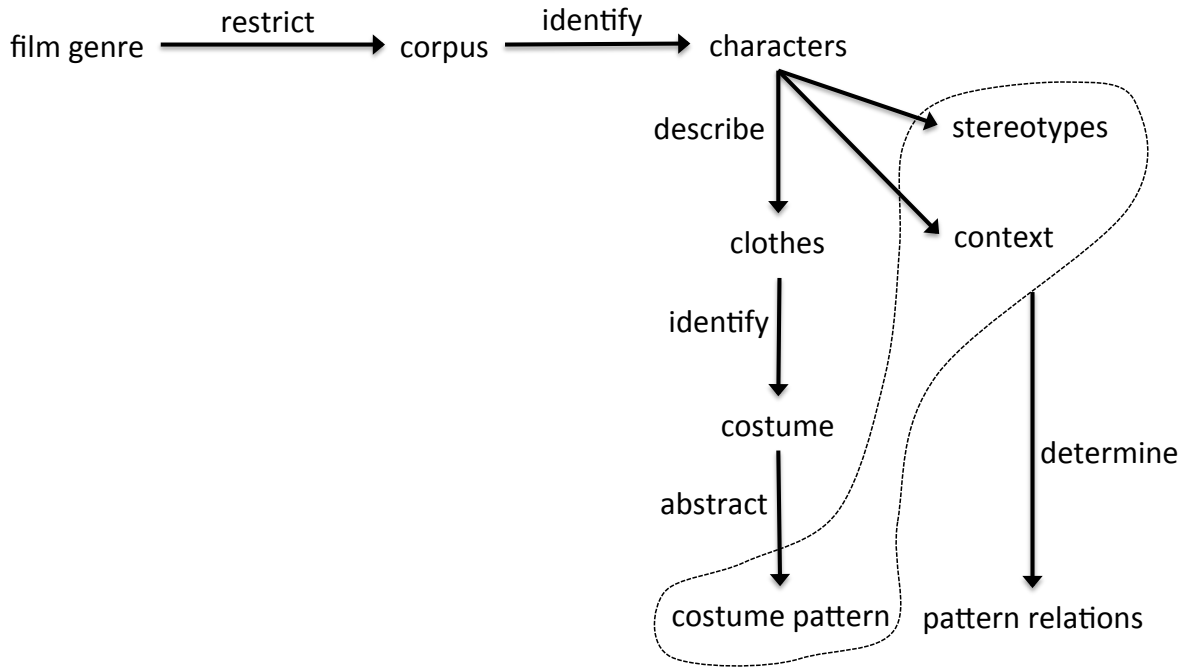

Figure 3: Method for the derivation of costume patterns

Costume patterns represent the “essence” of the convention of vestimentary communication. They are proven solutions to recurring design problems of finding an adequate textile expressions for a particular character. But patterns are not isolated entities but are related to each other. When considering the context in which the characters are set, relations between the patterns can be detected. Such relations could be, for example, “antagonist of” between the sheriff costume pattern and the outlaw costume pattern. All these patterns and their relations provide the *costume language* for the selected genre.

2.4. Combining Patterns and Solutions

When combining both the approach for describing clothes from section 2.2 and the approach for identifying costume patterns from section 2.3, a comprehensive method for deriving costume languages results (Figure 4). The method supports both, systematic pattern identification as well as the capturing of concrete solutions that led to the pattern abstraction. The concrete solutions (i.e. the costumes of the film corpus) and their relations are stored as instances of the ontology. This opens up new opportunities for working with costumes in theory and practice. By linking patterns with the ontology not only the knowledge contained in the pattern but also the information about concrete solutions are available. This can be very helpful when applying the pattern in a concrete context again (Falkenthal, Barzen, Breitenbücher, Fehling, & Leymann, 2014). The combination of patterns with their concrete solutions can be useful when working with costumes in practice: the use of advanced information technology for searching and documenting costumes is still rather uncommon (Schumm, Barzen, Leymann, Wieland, & Ellrich, 2012) and a system that supports our method (see section 4 and Fehling, Barzen, Falkenthal, & Leymann (2014)) is

very promising. This combination can also be used to extend theoretical knowledge about the application of costumes in films and can lead to new insights and better understandings about the nature of vestimentary communication.

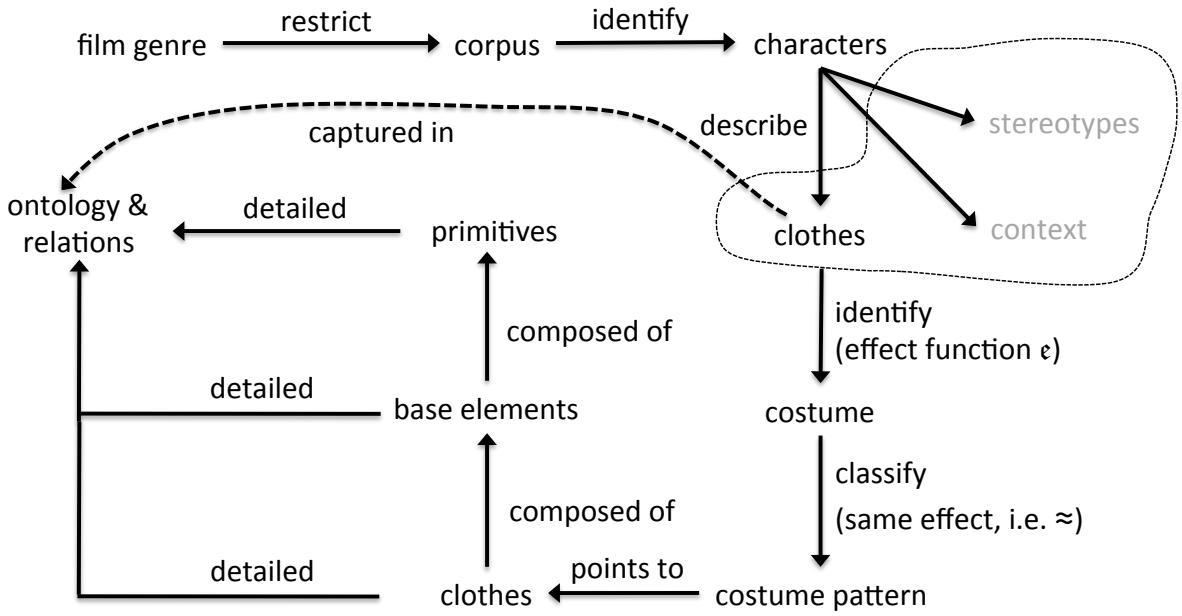

Figure 4: Method for deriving costume languages

Our method for detecting a pattern language for costumes is not only applicable in the domain of costumes but presents a first approach to systematic pattern identification in many domains (a first high level and domain independent approach is described in Fehling, Barzen, Breitenbücher, & Leymann, (2014)). In addition, it provides the opportunity to demonstrate how a pattern language has been derived. The patterns are no longer based on hidden knowledge of experts but become verifiable concept by making the original knowledge explicitly accessible: this supports pattern provenance.

Figure 5 presents our overall vision: Concrete solutions (be they observed or created etc.) are stored in a solution repository. This solution repository contains all the solution knowledge in a structured way (like the ontology in the domain of costumes). It gets analyzed and abstracted into patterns and pattern languages. The process of analysis and abstraction of solution knowledge can then be supported by query languages or other data analysis techniques such as data mining. The pattern languages are stored in a pattern repository, which allows people, for example, to navigate through the pattern languages or to search for concrete patterns. Furthermore, the connection of patterns to their associated concrete solutions in the solution repository makes the creation of concrete solutions from abstract solutions as documented by patterns often unnecessary or significantly reduces the effort needed to create new custom solutions. Thus, implementing abstract solutions as documented by patterns becomes much easier. Note explicitly that this vision does not restrict creativity at all. People can still create completely new solutions from scratch. An actual implementation to support our proposed method and vision is described in an accompanying article (Fehling, Barzen, Falkenthal, & Leymann, 2014).

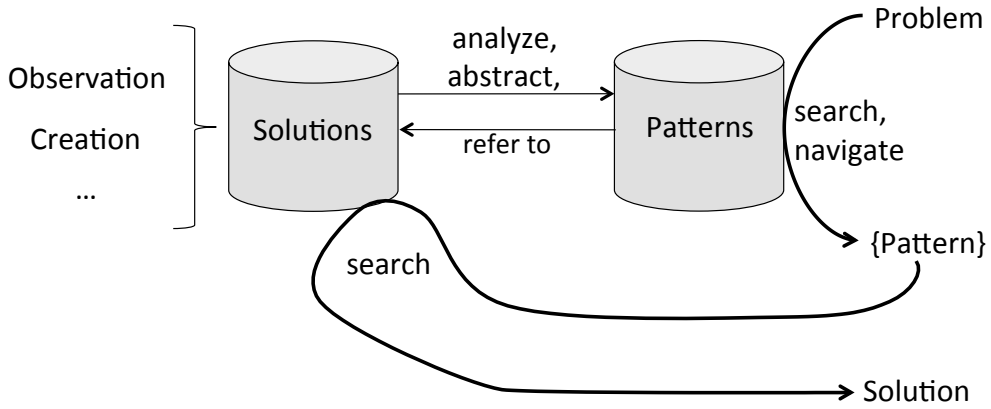

Figure 5: Method of the combined use of patterns and solutions

However, in order to approach the proposed detection of operationalized knowledge about costumes, a formalization of the costume language must be sought. Therefore, in a first step the formal definition of costume language is introduced which provides the basis for formalizing the pattern identification process. In the following section, the term "costume language" is concretized based on the concept of formal languages.

3. Concretization of the Term "Costume Language"

Clothes - in the sense of complete outfits such as the clothing of a sheriff – are, as already mentioned, composed of different individual pieces of clothes such as a shirt, a vest, a pair of trousers, a sheriff's star, a gun belt and boots. Each of these individual pieces are the building blocks of a *language of clothes* and are referred to as *base elements* (see Definition 1 and 5). The composition of clothes from base elements is determined by an appropriate grammar (see Definition 2) of base elements. By evaluating clothes with respect to their effect in films (see Definition 6) a selection of certain clothes results which is referred to as *costume fundus* (see Definition 7).

Costumes with similar effects in films are grouped and abstracted into *costume patterns* (see Definition 8). Thus, a costume is considered to be a proven solution to a re-occurring problem, namely the problem to achieve a particular effect, which is called a *pattern* in many other disciplines. Costumes in a film are typically not in isolation but they are related to each other, i.e. one particular costume appears often with other costumes. The structure of related costumes is then referred to a *costume language* (see Definition 9).

Base elements in turn are in general composed from (clothes) *primitives* such as a collar, a sleeve, a heel, a shoelace. The base element shirt, for example, consists of the primitives front, back, collar, cuffs and button border. Thus, the primitives are the building blocks of a *language of clothes primitives* (see Definition 10) determining the composition of base elements from primitives. A language of clothes primitives is especially relevant for manufacturing base elements, but of course clothes primitives may also contribute to the effect of the composed clothes.

3.1. Basic Terms

The *language of clothes* of a certain film genre is represented in the following as a formal language (Hopcroft & Ullman, 1969; Hopcroft, Motwani, & Ullman, 2007; Maurer, 1969). The alphabet of such a language of clothes consists of the base elements of the clothes that have been explicitly captured by analyzing a set of films (as described in Section 2); note, that our

terminology that is established in computer science deviates from everyday speech where an alphabet consists of a set of letters. The base elements (as well as the clothes primitives) are represented by concepts of the ontology of clothes and clothes primitives (see Figure 2). The alphabet of a language of clothes of a certain film genre is denoted by $\sum^B$ (Base Elements), which consists of the set of *base elements* of this film genre. For the genre of Western films this alphabet is $\sum^B$ ={Trousers, Shirt, Vest, Jacket, Sheriff-Star, ...}. In case a genre X must be emphasized in the following, we write $\sum^B(X)$.

As usual, we denote by $\sum^*$ the set of all possible words, i.e. composites from individual symbols from the alphabet $\sum$. Not all possible words are meaningful words, i.e., composites that are actually found in the domain: for example, (shoe boots moccasin) is a possible word over the alphabet $\sum^B$(Western), but this word does not make sense (i.e. it is not meaningful), because the combination of shoes, boots and moccasin does not occur in a single outfit.

Definition 1 (informally): A language L of clothes over the alphabet $\sum^B$ is the set of all "meaningful" words from $(\sum^B)^*$, i.e. $L \subseteq (\sum^B)^*$. ❒

To determine all "meaningful" words we use *grammars* that define by means of *production rules* the composition of clothes from base elements.

Definition 2: A g*rammar* G is a tuple G=(V, $\sum$, P, S) with:

- $\sum$ is an alphabet
- V is a *vocabulary* (with $\sum \subseteq V$)
- P is the set of production rules over V
- S is the *start symbol* (with $S \in V \setminus \sum$). ❒

The elements from $V \setminus \sum$ are called *variables*: a variable is a name of composite. For example, the variable suit is composed from trousers and jacket, the latter are base elements from the alphabet $\sum$. That is, a corresponding language of clothes would have a production rule that describes this combination (see section 3.2. that provides a more comprehensive example).

Definition 3: A *production rule* is a pair of words over the vocabulary V of a grammar G=(V, $\sum$, P, S) such that:

$$(x,y) \in P :\Leftrightarrow x \in V^* \setminus \sum^* \text{ und } y \in V^*.$$

Instead of $(x,y) \in P$ we write: $x \rightarrow y$. ❒

3.2. Sample Language of Clothes of the Genre of "Western Movies"

The following example sketches the language of clothes of the genre of Western movies. The corresponding sample alphabet $\sum$'={Trousers, Jacket, Skirt, Loincloth, Vest, Corsage, Scarf, Boots, Ankle-Boots, Moccasins, Shoes, Shirt, Blouse, Hat, Headband, Feather-Trimming, Sheriff-Star} is the set of base elements of the genre. The set {Sheriff, Barman, Saloon-Lady, Indian, Bandit, Suit} is the set of sample variables; these variables are names for possible clothes of roles in Western films. Thus, the overall vocabulary of our example is V'={Trousers, Jacket, Skirt, Loincloth, Vest, Suit, Corsage, Scarf, Boots, Ankle-Boots, Moccasins, Shoes, Shirt, Blouse, Hat, Headband, Feather-Trimming, Sheriff-Star, Sheriff, Barman, Saloon-Lady, Indian, Bandit}.

The following rules are a sample set of production rules P' over the vocabulary V' (S is the start symbol):

(1) Suit ⟶ Trousers Vest Jacket
(2) Suit ⟶ Trousers Jacket
(3) Sheriff ⟶ Suit Shirt Boots Sheriff-Star
(4) Sheriff ⟶ Trousers Vest Boots Shirt Sheriff-Star
(5) Sheriff ⟶ Trousers Shirt Boots Sheriff-Star
(6) Sheriff ⟶ Trousers Vest Boots Shirt Hat Sheriff-Star
(7) Sheriff ⟶ Trousers Shirt Boots Hat Sheriff-Star
(8) Barman ⟶ Trousers Vest Shoes
(9) Saloon-Lady ⟶ Skirt Blouse Corsage Ankle-Boots
(10) Saloon-Lady ⟶ Skirt Corsage Ankle-Boots
(11) Saloon-Lady ⟶ Skirt Blouse Corsage Ankle-Boots Hairband
(12) Saloon-Lady ⟶ Skirt Corsage Ankle-Boots Hairband
(13) Indian ⟶ Moccasins Loincloth Vest Feather-Trimming
(14) Indian ⟶ Moccasins Loincloth Vest
(15) Indian ⟶ Moccasins Loincloth Feather-Trimming
(16) Indian ⟶ Loincloth Feather-Trimming
(17) Bandit ⟶ Trousers Shirt Vest Scarf Boots Hat
(18) Bandit ⟶ Suit Shirt Scarf Boots Hat
(19) S ⟶ Sheriff
(20) S ⟶ Barman
(21) S ⟶ Saloon-Lady
(22) S ⟶ Indian
(23) S ⟶ Bandit

These production rules generally result from an analysis of a corpus of films that has been selected as being representative for a certain genre. After analysis of the corpus all production rules have been captured that specify the composition of clothes from base elements of the roles that occur in the analyzed films. Thus, the clothes of the genre are described words over the alphabet $\sum$ of the base elements, which can be produced by these production rules. As usual, *produce* means – even repeated – application of production rules, i.e. the replacement of a symbol in a word on the left side of a production rule by the word of the right side of the production rule. If a word y can be produced from another word x by applying production rules, it is usual to write $x \rightsquigarrow y$.

Applying production rules (19), (3) and (1) yields $S \rightsquigarrow$ (Trousers Vest Jacket Shirt Boots Sheriff-Star), i.e. this word represents valid clothes in a Western film:

S ⟶ Sheriff ⟶ Suit Shirt Boots Sheriff-Star
⟶ Trousers Vest Jacket Shirt Boots Sheriff-Star

The grammar G'=(V',∑',P',S) is an example of how the meaningful words, i.e. clothes, of the language of clothes of Western films can be determined.

3.3. Language of Clothes and Costume Fundus

The set of all words of a grammar that can be generated from the start symbol S, is also called "the language generated by the grammar." A language generated by a grammar is thus the set of all meaningful words that can be build by composing symbols from an alphabet ∑; hereby, the term "meaningful", which was still used informally in Definition 1, is clarified.

Definition 4: The set of all words that can be generated from the start symbol and the production rules of a grammar G is called the *language L(G) generated by the grammar G*. Thus, it is

$$L(G) := \{x \in \sum^* \mid S \rightsquigarrow x\} \quad \square$$

For our sample grammar G'=(V',∑',P',S) from Section 3.2, L (G') is the set of all clothes of our sample corpus of Western movies.

To define the language of clothes of a genre the vocabulary of this genre must be defined first. For this purpose, the base elements $\sum^B$ of the genre have to be identified and the variables $V^B \backslash \sum^B$ have to be specified; in practice, this is done by analyzing the corpus of the corresponding genre. Similarly, the production rules P^B are determined by analysis of the corpus. The choice of an arbitrary start symbol S^B then completes the grammar $G^B=(V^B,\sum^B,P^B,S^B)$ of the language of clothes of the genre. The language of clothes of the genre itself then is the language $L(G^B)$ generated by the grammar G^B:

Definition 5: The language $L(G^B)$ generated by the grammar $G^B=(V^B,\sum^B,P^B,S^B)$ is called *language of clothes*. ❐

This definition of a language of clothes goes far beyond the domain of movies, i.e., the above formal definition is applicable in a variety of domains in which clothes play a role. For example, clothes with a specific use such as casual wear, protective clothes, festive clothing, etc. can be considered as languages of clothes. Similarly, clothes that have been created by a certain fashion designer or clothes offered by a particular fashion label may be seen as separate languages of clothes. Further applications of the above definition of languages of clothes such as clothes worn by actors in theater productions or clothes of subcultures like punks or skinheads seem to be obvious.

In the domain we are mostly interested in in this contribution, namely clothes that actors wear in movies, clothes is referred to as *costume.* I.e. the plain use of clothes in a movie turns clothes into costumes (aside: the same is true for clothes used in the theater). In this sense, in the domain of movies one is tempted to speak interchangeably of costume language and language of clothes. But this synonymous use of "clothes" and "costume" in the domain of movies (and, thus, equating the terms "language of clothes" and "costume language") falls short: a costume is clothes with intended effect in a movie (see Definition 6 and Definition 7). Also, a language of clothes has to capture the conventions of the use of clothes, i.e. the recurring use of clothes to solve the problem of achieving particular effects in movies. Typically, a variety of clothes achieve the same effect, i.e. the formation of equivalence classes (of clothes achieving the same effect) is necessary to turn concrete clothes into elements of a costume language (see Definition 8).

Thus, concrete clothes that are observed in movies must first be analyzed for their effect. The explicit choice of clothes with intended effect from the set of words of a language of clothes $L(G^B)$ results in a set of costumes that we call *costume fundus* $\mathfrak{C}$; a costume fundus $\mathfrak{C}$ is a subset of a language of clothes $L(G^B)$, i.e. $\mathfrak{C} \subseteq L(G^B)$. The (intended) effect of clothes can be determined in various ways. It can be found, for example, by having a representative test group of spectators or a set of movie experts analyzing the movie corpus. Or the frequency of occurrence of certain clothes in the corpus can be taken as evidence of the effect as a costume, at least signaling the convention of the use of the corresponding clothes. Formally, we represent the selection of clothes from $L(G^B)$ because of their effect by a Boolean function $\mathfrak{e}$:

Definition 6: A function $\mathfrak{e}$: $L(G^B) \longrightarrow$ {true, false} is called *effect function*. ❒

Thus, a costume c is a piece of clothes $c \in L(G^B)$ "with effect", i.e. a piece of clothes c becomes a costume if and only if $\mathfrak{e}(c)$=true:

Definition 7: The set $\mathfrak{C} = \{c \in L(G^B) \mid \mathfrak{e}(c)=\text{true}\}$ is called *costume fundus*. ❒

3.4. Costumes as Patterns and the Definition of Costume Language

The effect function $\mathfrak{e}$ indicates whether or not a piece of clothes c has an effect or not, i.e. whether a piece of clothes is a costume of not. A costume language needs to go beyond just indicating that a piece of clothes has an effect: it must express this effect by showing the conventions behind the costumes. In order to express conventions behind the use of costumes, costumes with "the same effect" must be identified. This identification of costumes that have the same effect is formalized by an equivalence relation ≈ (see Definition 8). For a costume c its equivalence class $[c]_{\approx}$ is the set of all costumes with an effect similar to that of c (see Definition 8).

Example: In the Western genre, all clothes worn by a sheriff in one of the movies of the corpus are analyzed with respect to their effect. Some of these clothes may not have the effect of signaling of "being a sheriff", e.g. when a sheriff wears pajamas because he is about going to bed. This kind of clothes is omitted from the costume fundus because the effect function $\mathfrak{e}$ will assign to these clothes the value "false". In the Western genre, there will be other costumes like the ones of barmen etc.: the equivalence relation ≈ will separate all costumes of sheriffs from all costumes of a barmen by putting them into separate equivalence classes $[\text{sheriff}]_{\approx}$ and $[\text{barman}]_{\approx}$.

Each such equivalence class $[x]_{\approx}$ represents the proven solutions of the problem of achieving a particular effect in a movie by using certain clothes. E.g. the equivalence class $[\text{sheriff}]_{\approx}$ contains all sheriff costumes that have been successfully used in the movies of the corpus to signal that a character is a sheriff, i.e. each one of these costumes may be used in future to indicate a sheriff. Such proven solutions to recurring problems are called *patterns* (Alexander et al., 1977).

Typically, solutions are considered "proven" if they have been successfully applied multiple times. Thus, we define a natural number $N \in \mathbb{N}$ such that only equivalence classes with $\text{card}([x]_{\approx})>N$, i.e. equivalence classes with more than N members, are considered as patterns. We call such patterns in our domain *costume patterns*. Thus, a certain costume pattern consists of a set of more than N costumes that achieve the same effect.

Definition 8: The equivalence relation $\approx \subseteq \mathfrak{C}\times\mathfrak{C}$ determines costumes with the same effect, i.e.

$$(x,y) \in \approx \ :\Leftrightarrow \ \text{x and y have “the same effect”.}$$

Instead of $(x,y)\in\approx$ it is also usual to write $x\approx y$.

$[x]_\approx := \{y\in\mathfrak{C} \mid x\approx y\}$ is an equivalence class of costumes with the same effect. The set of all equivalence classes is denoted by

$$\mathfrak{P} := \mathfrak{C}/\approx = \{[x]_\approx \mid x\in\mathfrak{C}\}.$$

With $N\in\mathbb{N}$ we call

$$\mathfrak{P}_N := \{[x]_\approx \mid x\in\mathfrak{C} \wedge \text{card}([x]_\approx)>N\}$$

the set of *costume patterns*. ❒

Figure 6 depicts the relation between costumes and costume patterns: costumes from the costume fundus $\mathfrak{C}$ that have the same effect are abstracted by means of the equivalence relation $\approx$ into a costume pattern $P_i \in \mathfrak{P}$. Thus, identifying costumes that solve the same effect problem, i.e. that have the same effect, is a process of abstraction.

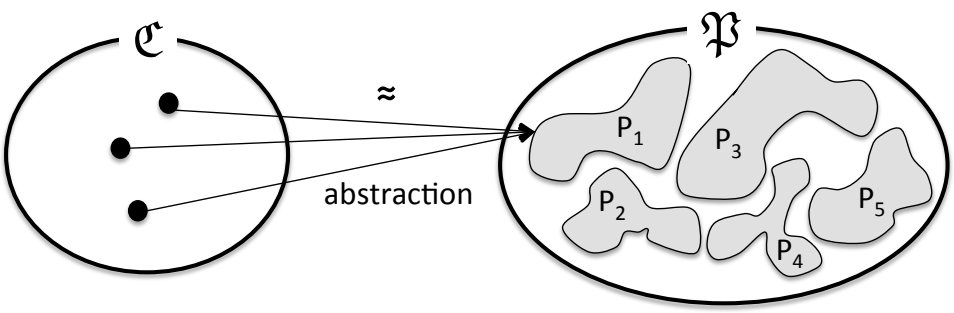

Figure 6: Identification of patterns as a process of abstraction

Figure 7 gives a concrete example of this abstraction process: the three costumes of a sheriff in the costume fundus $\mathfrak{C}$ are abstracted into the “sheriff” costume pattern shown in the set of costume patterns $\mathfrak{P}$.

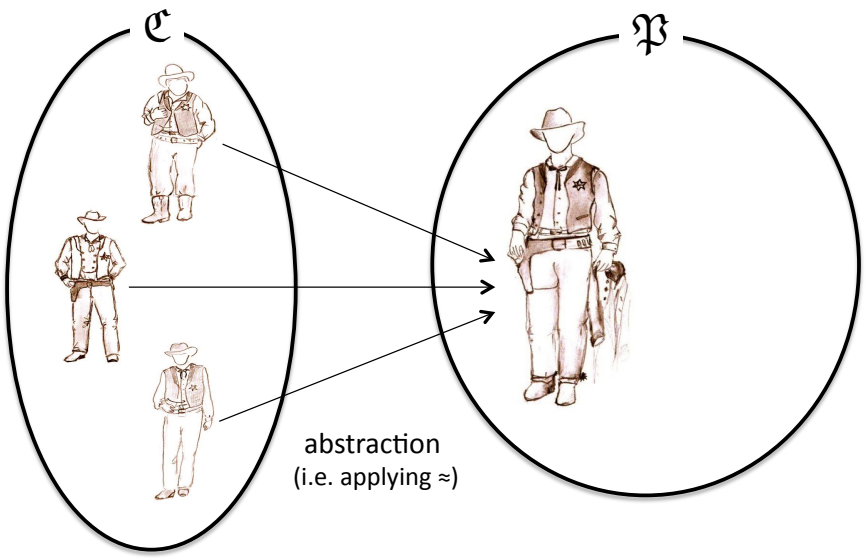

Figure 7: Sample abstraction of costumes into a pattern

Costume patterns are not isolated but they are related to each other. This is because in situations in which a certain pattern is applicable other patterns are often applicable too. Such relations are described by directed edges pointing from one pattern to another one. These relations often have certain semantics: for example, the Barman costume pattern often "jointly appears" with the Saloon-Lady costume pattern, or the Sheriff costume pattern is often related by "fondness" to the Saloon-Lady costume pattern (see Figure 8).

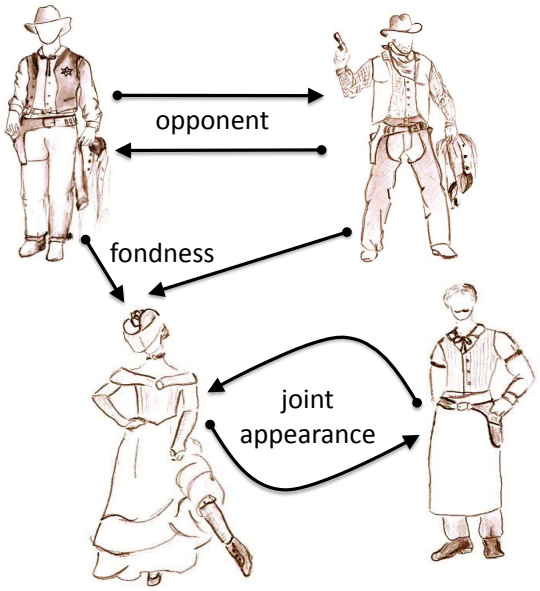

Figure 8: Sample costume language

Thus, costume patterns together with the pattern relations form a directed graph (see Figure 9) is called costume language.

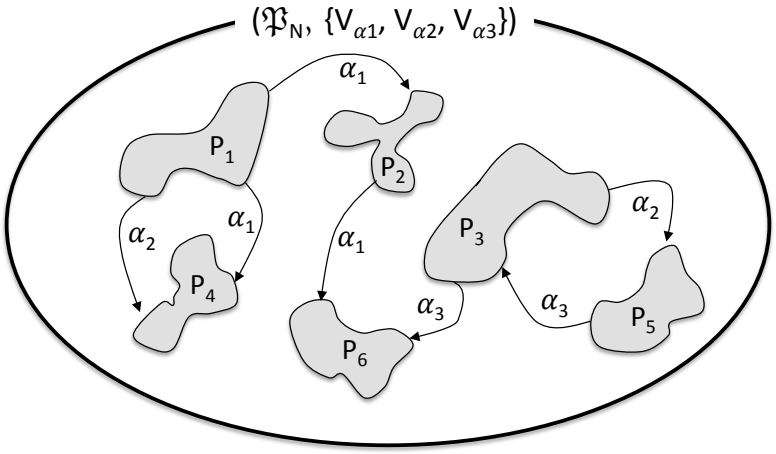

Figure 9: Costume language as a graph of costume patterns

Definition 9: Let $V_\alpha \subseteq \mathfrak{P}_N \times \mathfrak{P}_N$ be the set of all relations of a certain semantics $\alpha \in \{\alpha_1,...,\alpha_k\}$ between costume patterns. $\{V_\alpha\}$ is the set of *pattern relations* between the costume patterns $\mathfrak{P}_N$. The pair ($\mathfrak{P}_N$, $\{V_\alpha\}$) is called *costume language*. ❒

The justification for calling this graph a "language" is as follows: to solve a particular problem a first pattern of the pattern language is identified. Then the relations of this pattern to other patterns are inspected. If appropriate some of these patterns are selected to "enhance" or "complete" the first solution and so on. This way, combinations of patterns are built similar to building words from an alphabet. The difference to the notion of a language from Definition 5 is that meaningful words are not generated by a grammar but by navigating through a graph. Nevertheless, the graph determines what meaningful combinations are.

3.5. Language of Clothes Primitives and Operational Aspects of Clothes

The composition of base elements from clothes primitives is also of importance to determine costumes (and their effect): the difference between a leather vest without a collar and a fabric vest with shawl collar as a concrete realization of the base element "vest" is essential. Therefore, the set $\sum^p$ (primitives) of clothes primitives of the costumes of the genre and their composition into basic elements is determined. For example, the alphabet of clothes primitives for the genre of Western movies is $\sum^p$ = {collar, sleeves, ...}. The specific composition of the base elements from primitives is described by a set of production rules P^p. Thus, we can define a grammar G^p the variables of which are the base elements, the alphabet of which are the clothes primitives, and the production rules describe the compositions of base elements from primitives.

Definition 10: The language $L(G^p)$ generated by the grammar $G^p=(V^p,\sum^p,P^p,S^p)$ is called *language of (clothes) primitives*. ❒

$L(G^p)$ is the language that describes the composition of base elements from clothes primitives. This language is mostly of "technical" interest, such as for sewers, dressers, etc. Its aspect that is relevant for a costume fundus is in describing subtle differences in the base elements which in turn may correlate e.g. with special characteristics of a role. However, the language of primitives is an important aspect of a costume language as a pattern language, i.e. as a language proving proven solutions to recurring problems: the language of primitives specifies the "manufacturing" part of the overall solution of the "effect" problem to be solved by a costume.

Thus, our approach connects the base elements of a costume $c \in \mathfrak{C}$ with words from the language of clothes primitives $L(G^p)$ to describe the composition of base elements of a costume from primitives: let c be a costume, i.e. c is a word $b_1...b_n$ from $L(G^B)$. Then, each base element $b_i \in \sum^B$ of the word c can be associated with a word from $L(G^p)$. This word $p_{i1}...p_{ik}$ from $L(G^p)$ describes the composition of the base element b_i of the costume from primitives $p_{i1}...p_{ik}$. By substituting each base element b_i of the costume $c=b_1...b_n$ by the corresponding word $b_i=p_{i1}...p_{ik}$ we derive $c=p_{11}p_{12}...p_{i1}p_{i2}...p_{nk}$: thus, the costume is described in its composition from primitives, i.e. the costume became a subject of manufacturing. As a consequence, the costume can be (re-)produced.

Beside understanding the production of each base elements of a costume the correct combination of these base elements into the costume to solve an effect problem is relevant. If the costume of the sheriff contains, among other things, a shirt, a vest and a sheriff's star, it is essential whether the sheriff's star is attached to the shirt or to the vest. In our approach relations between the base elements represent this information.

Definition 11: A subset of pairs of base elements with a certain meaning $\mathfrak{B}$ is called *base relation* $R_{\mathfrak{B}}$:

$$R_{\mathfrak{B}} \subseteq \sum^B \times \sum^B. \text{ ❒}$$

In this context, "meaning" $\mathfrak{B}$ states, for example, whether for a pair $(x,y) \in \sum^B \times \sum^B$ it is true that x is fixed to y (i.e., $\mathfrak{B}$ = fixed_to – e.g. Sheriff's star fixed to vest); or that x is worn over y (i.e. $\mathfrak{B}$ = worn_over – e.g. vest worn over shirt); or x is wrapped around y (i.e. $\mathfrak{B}$ = wrapped_around – e.g. Belt wrapped around trousers); etc.

These base relations are thus an essential aspect of a costume as a solution to a problem: they specify an operational way to combine base elements into a costume (Schumm, Barzen, Leymann, & Ellrich, 2012). Because of their operational semantics we call base relations also *(combination) operators*. The operators and their order of application imply a procedure, which is essential for dressing of a costume (Schumm, Barzen, Leymann, & Ellrich, 2012).

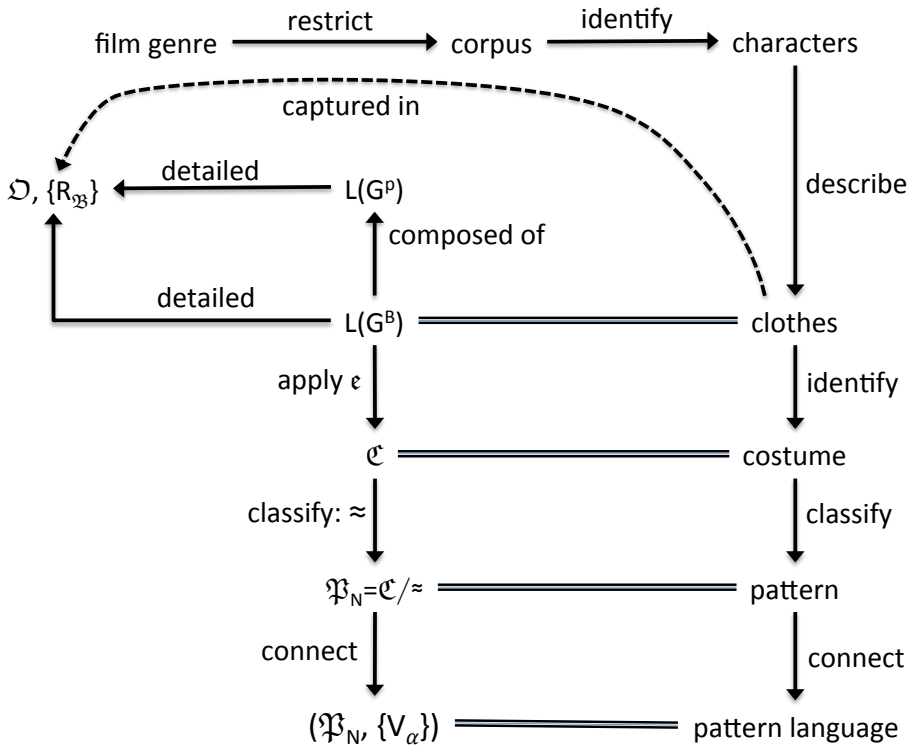

Figure 10: Formal concepts within our method to derive costume languages

By refining Figure 4 with the formal concepts developed in this section, Figure 10 summarizes the formal aspects of our approach and relates them to the methodological aspects of our approach from section 2. The ontology used for capturing details about clothes is denoted by $\mathfrak{O}$, while the set of base relations, which describe the operational semantics of the combination of the base elements is denoted by $\{R_{\mathfrak{B}}\}$. Clothes primitives (i.e. $L(G^P)$) are captured in the ontology $\mathfrak{O}$, and the same is true for base elements (i.e. $L(G^B)$). Clothes correspond to words of $L(G^B)$ as indicated by the double-line in the figure. By applying the effect function $\mathfrak{e}$ to clothes, the costume fundus $\mathfrak{C}$ results. Grouping costumes that have the same effect, i.e. applying the equivalence relation ≈, and considering only "proven" solutions (i.e. ones that appear more than N times), results in the set of costume patterns $\mathfrak{P}_N$. Connecting the patterns with the observed relations between them results in the costume language.

4. Generalization: Formalism for Solutions and Pattern Languages

Our formalization can be generalized to be applicable in other domains in which solutions should be abstracted to patterns and structured into pattern languages. The generalized formalization (see Figure 11) is a first step towards a formal joint representation of solutions and patterns, as well as the relation between solutions and pattern languages.

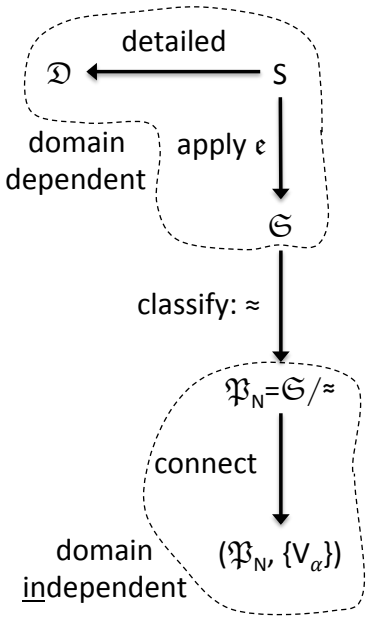

Figure 11: Formal concepts for pattern languages with associated solutions

In another domain, the ontology $\mathfrak{O}$ as well as the relations $\{R_{\mathfrak{B}}\}$ between the corresponding concepts of $\mathfrak{O}$ must to be substituted by another model $\mathfrak{D}$ of the corresponding universe of discourse (see Figure 11). The domain specific model $\mathfrak{D}$ captures the metadata (types) about this domain as well as the data (instances), just like $\mathfrak{O}$ captures the types of clothes (e.g. base elements like trousers, shirts, belts, boots,…) and their relations as well as all concrete clothes observed in the corpus. In another domain, a formal language S ("Solution Language") is used to control all valid occurrences of combinations of the types (and their instances), just like $L(G^B)$ controls all valid combinations of base elements into clothes. $L(G^P)$ represents additional details about the base elements which is irrelevant for the analogy with other domains here and is, thus, omitted – but other domains may similarly refine their solution language.

Next, words from S must be evaluated with respect to their effect, i.e. whether or not they achieve the goal of being a solution to a problem in the domain. Thus, a domain specific effect function $\mathfrak{e}$ is applied to derive the set of effective solutions $\mathfrak{S}$, just like the costume fundus $\mathfrak{C}$ is derived. By identifying solutions to the same problem, i.e. by treating such solutions as "equivalent" by means of a domain specific equivalence relation ≈, the patterns $\mathfrak{P}_N$ of the domain are identified (only considering equivalence classes above a certain cardinality to indicate that the class of solutions are proven). Finally, patterns are associated based on the domain specific set of relations $\{V_\alpha\}$. This way, the pattern language $(\mathfrak{P}_N, \{V_\alpha\})$ of the domain is created.

The domain model $\mathfrak{D}$, the solution language S and, consequently, the set of effective solutions $\mathfrak{S}$ is domain dependent; the same is true for the effect function $\mathfrak{e}$ and the equivalence relation ≈. Thus, for each domain in which pattern languages should be derived from solutions or new working solutions associated with a patterns should be documented, the formal concepts $\mathfrak{D}$, S, $\mathfrak{e}$, $\mathfrak{S}$ and ≈ must be specified. In contrast, the way in which patterns $\mathfrak{P}_N$ are described and related (i.e. $\{V_\alpha\}$) is (nearly) domain independent (note, that the semantics of the relations may be domain dependent).

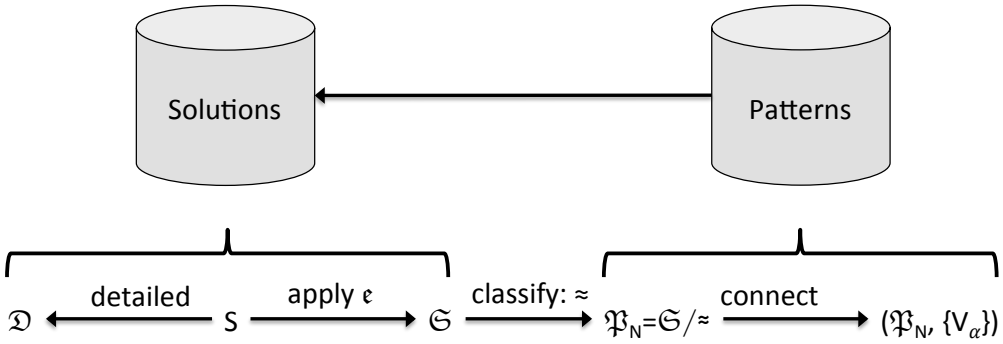

Figure 12: Implementation aspect

The knowledge about solutions and patterns from a specific domain can be stored in corresponding repositories. *Solution repositories*, i.e. repositories that store knowledge about solutions, are highly dependent on the domain model $\mathfrak{D}$ as well as the solution language S as they define the metadata of the domain and control valid combinations. Thus, solution repositories must be built for each domain specific to $\mathfrak{D}$ and S (see Figure 12). The kind in which patterns are described turned out to be quite domain independent, i.e. a single *pattern repository* seem to suffice to capture pattern languages ($\mathfrak{P}_N$, $\{V_\alpha\}$) from very different domains. Finally, links between patterns and the solutions they have been derived from (or solutions that are build according to a pattern's high-level solution description and then "harvested" into the solution repository) can be established based on a multitude of linking mechanisms (e.g. RDF).

5. Summary and Outlook

In this article, we proposed a method to identify costume patterns in the domain of movies as well as a corresponding pattern language. This method is also applicable to other domains. Furthermore, we have formally defined the term "costume language". This formalization can be used for both purposes, capture operational knowledge about costumes as well as provide a solid base for discussions about costumes in the media science. Our method and formalization may also be seen as a contribution to Digital Humanities proving that modeling, formal languages, and patterns are powerful tools in these domains too.

Currently, we are in the process of capturing clothes from a corpus of movies. The captured information is based on the formalization of clothes and costumes described in this article, and it is stored in a corresponding solution repository. The solution repository is analyzed to derive the costume patterns and relations between them and the result is stored in a pattern repository (see Figure 13); the repositories follow the architecture from Figure 12. Details about these repositories and the corresponding tool environment are described in the accompanying article (Fehling, Barzen, Falkenthal, & Leymann, 2014).

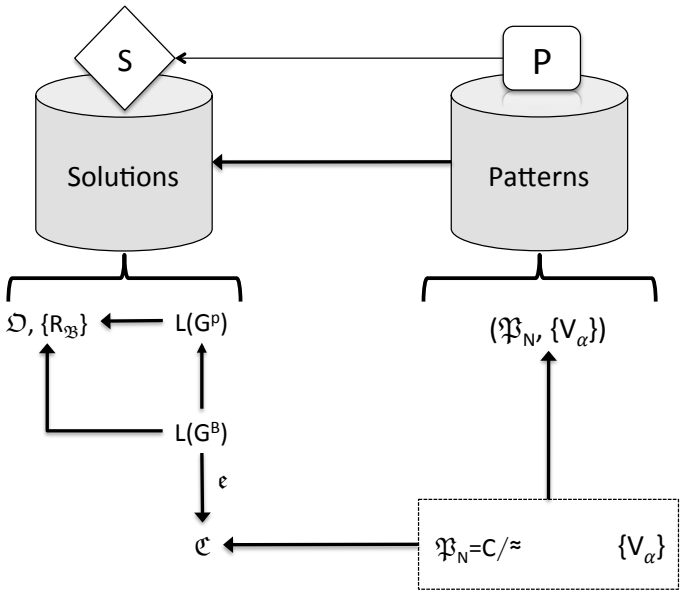

Figure 13: Formal concepts and repositories

6. Acknowledgments

We are grateful to Christoph Fehling for many discussions on pattern languages and their methodological determination. Also, we would like to thank Michael Falkenthal for the discussions on the feasibility of solution repositories and his insight in their implementations. Furthermore, we want to thank all participants of the PURPLSOC Workshop for their feedback, good ideas and support, and special thanks to the participants of our working group.

7. References

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language – Towns Buildings Construction*. Oxford: Oxford University Press.

Barthes, R. (1985). *Die Sprache der Mode* [Système de la Mode 1967]. Frankfurt/M.: Suhrkamp.

Bast, H. (2013). Semantisch Suche. *Informatik Spektrum, 36*(2), 136-143.

Barzen, J. (2013). Taxonomien kostümrelevanter Parameter: Annäherung an eine Ontologisierung der Domäne des Filmkostüms. *Technischer Bericht Nr. 2013/04* (Universität Stuttgart). Retrieved from http://www2.informatik.uni-stuttgart.de/cgi-bin/NCSTRL/NCSTRL_view.pl?id=TR-2013-04&mod=0&engl=0&inst=FAK. (accessed on 7 August 2014)

Barzen, J., Leymann, F., Schumm, D., & Wieland, M. (2012). Ein Ansatz zur Unterstützung des Kostümmanagements im Film auf Basis einer Mustersprache. *Proceedings Modellierung 2012*, 251-266.

Berry, D. M. (Hrsg.). (2012). *Understanding Digital Humanities*. Basingstoke: Palgrave Macmillan.

Bruzzi, S. (1997). *Undressing Cinema. Clothing and Identity in the movies*. Abingdon, Oxford: Routledge.

Burdik, A., Drucker, J., Lunenfeld, P., Presner, T., & Schnapp, J. (2012). *Digital_Humanities*. Cambridge/London: MIT Press.

Burger, R. (2002). *Contemporary Costume Design. Dress Codes und weibliche Stereotype im Hollywood-Film*. Wien: Facultas.

Cook, P. (1996). *Fashioning the Nation. Costume and Identity in British Cinema*. London: British Film Institute.

Devoucoux, D. (2007). *Mode im Film. Zur Kulturanthropologie zweier Medien*. Bielefeld: Transcript.

Falkenthal, M., Barzen, J., Breitenbücher, U., Fehling, C., & Leymann, F. (2014). From Pattern Languages to Solution Implementations. *Proceedings of the Sixth International Conferences on Pervasive Patterns and Applications (PATTERNS 2014)*.

Fehling, C., Barzen, J., Breitenbücher, U., & Leymann, F. (2014): The Process of Pattern Identification, Extraction, and Application. *Proceedings of the 19th European Conference on Pattern Languages of Programs (EuroPLoP)*.

Fehling, C., Barzen, J., Falkenthal, M., & Leymann, F. (2014). PatternPedia. Collaborative Pattern Identification and Authoring. In: Proc. of Preparatory Workshop of PURPLESOC 2015.

Fehling, C., Leymann, F., Retter, R., Schupeck, W., & Arbitter, P. (2014) *Cloud Computing Patterns*. Wien/Heidelberg/New York: Springer.

Giannone, A. (*2005*). *Kleidung als Zeichen. Ihre Funktionen im Alltag und ihre Rolle im Film westlicher Gesellschaften: eine kultursemiotische Abhandlung*. Berlin: *Weidler.*

Hey, T., Tansley, S., & Tolle, K. (2009). *The Fourth Paradigm - Data-intensive Scientific Discovery*. Redmond, Washington: Microsoft Corporation.

Hollander, A. (1993). *Seeing through Clothes*. Berkeley/Los Angeles/London: University of California Press.

Hopcroft, J.E., & Ullman, J.D. (1969). *Formal Languages and Their Relation to Automata*. London: Addison-Wesley.

Hopcroft, J.E., Motwani, R., & Ullman, J.D. (2007). *Introduction to Automata Theory, Languages, and Computation* (3rd Edition). London: Pearson.

Lurie, A. (2000). *The Language of Clothes. Featuring a new introduction about fashion today*. New York: Owl Books.

Maurer, H. (1969). *Theoretische Grundlagen der Programmiersprachen – Theorie der Syntax*. Berlin: Bibliographisches Institut.

McDonald, T. J. (2010). *Hollywood Catwalk. Exploring Costume and Transformation in American Film*. London/New York: I.B. Tauris.

Schumm, D., Barzen, J., Leymann, F., Wieland, M., & Ellrich, L. (2012). Business Process Automation for Costume Management in Film Making: An Insight into Processes, Roles, and Document Structures. *EMISA Forum. Vol. 32(1)*, Gesellschaft für Informatik e.V. (GI), 26-47.

Schumm, D., Barzen, J., Leymann, F., & Ellrich, L. (2012). A Pattern Language for Costumes in Films. *Proceedings of the 17th European Conference on Pattern Languages of Programs (EuroPLoP)*, 221-250.

Taibi, T. (2006). Formalising Design Patterns Composition. *IEE Proc.-Softw., 153(3)*, 127 - 136.

Terras, M., Nyhan, J., & Vanhoutte, E. (2013). *Defining Digital Humanities. A Reader*. Farnham: Ashgate Publishing Limited.

Zdun, U. (2007). Systematic Pattern Selection Using Pattern Language Grammars and Design Space Analysis. *Software: Practice & Experience, 37(9),* 983 - 1016.

Zdun, U., & Avgeriou, P. (2008). A Catalog of Architectural Primitives for Modeling Architectural Patterns. *Information and Software Technology, 50 (9-10)*, 1003-1034.

8. About the author/s:

Johanna BARZEN studied media science, musicology and phonetics at the University of Cologne and gained first practical experience while working for some major television channels like WDR and RTL. Next to this she studied costume design at the ifs (international film school Cologne) and worked in several film productions in the costume department in different roles. Currently she is Ph.D. student at the University of Cologne and research staff member at the Institute of Architecture of Application Systems (IAAS) at the University Stuttgart doing research on vestimentary communication in film.

Frank LEYMANN is a full professor of computer science and director of the Institute of Architecture of Application Systems (IAAS) at the University of Stuttgart, Germany. His research interests include service-oriented architectures and associated middleware, workflow- and business process management, cloud computing and associated systems management aspects, and patterns. Frank is co-author of more than 300 peer-reviewed papers, more than 40 patents, and several industry standards. He is on the Palsberg list of Computer Scientists with highest h-index.