JPMTR-2310
DOI 10.14622/JPMTR-2310
UDC 004.891:004.423:025.352

Research paper | 185
Received: 2023-08-21
Accepted: 2023-12-26

TaxoCatalog: expert system for semantically personalizing paper-based product catalogs in omni-channel context using background knowledge

Heiko Angermann

Hochschule Darmstadt, Faculty of Design,
Olbrichweg 10, 64287 Darmstadt, Germany

heiko.angermann@h-da.de

Abstract

Taxonomies are a formal method of semantically structuring information using hierarchically ordered concepts. Those play a crucial role in omni-channel retailing to publish product catalogs across media channels, i.e. digital (portal-based, paper-based) media channels, and print media (paper-based) media channels. Portal-based media channels use taxonomies to structure product-related content by concepts to facilitate customers navigate through the e-commerce site. That are the product categories. Paper-based media channels require taxonomies for automatic layout setting using data-driven publishing software. When recommender systems are additionally used, products are published individually on the e-commerce site. Printed product catalogs, on the other hand, currently only display content regardless of dynamic preferences, unless the layout is set manually. That is, as in an industrial context, personalization is of no interest if the necessary processes cannot be automated. The only recent industrially relevant method of taking preferences into account is to print sub-catalogs. However, these only contain certain product categories, resulting in a loss of information and sales, as preferences change dynamically today. With TaxoCatalog, an expert system is presented in this paper, capable of semantically personalizing product taxonomies to layout printed product catalogs according to the dynamic preferences of customers. The proposed expert system considers three layers to achieve full automation of relevant processes. The first layer uses background knowledge to consider a memory-based analysis of preferences, and a content-based analysis of possible semantic modifications. The second layer infers semantically personalized taxonomies using different modification rules. The third layer transforms the individual taxonomy paths into XML. This allows that the output of TaxoCatalog can be processed by any standard data-driven publishing software for automatic layout setting. A case study and a comprehensive evaluation, which discusses the strengths and limitations of previous research in the field, as well as the expert system in terms of quantitative and qualitative criteria, underline the efficiency of TaxoCatalog.

Keywords: data-driven publishing, expert systems, page layout, cross-channel publishing, taxonomy

1. Introduction

Today, products are advertised on different media channels (Hänninen, Kwan and Mitronen, 2021). Formally, this is referred to as multi-channel retailing strategy, or as omni-channel retailing strategy when the different channels offered are connected to each other in one retail network (Herhausen, et al., 2019; Beck and Rygl, 2015). Digital media channels can be used in the form of internal and external operated sales portals (e-commerce, marketplaces), or in the form of paper-based digital documents. Paper-based printed media products can be used as documents (Gao, Melero and Sese, 2020). All have their own advantages, but when using e-commerce, retailers can infer customer preferences in real time using recommender systems (Chen, et al., 2023). As a result, products can be individually displayed to customers based on their likelihood of ordering (Behera, et al., 2020). For example, suitable products are shown at the bottom of the product site, known as cross- and up-selling (Norvell, Kumar and Contractor, 2018). Paper-based digital product catalogs, on the other hand, display static content, even if the appearance is dynamic. Here, formats like EPUB, or (interactive) PDF are used. Printed product catalogs, also show static content, as individualization affects the de-automation of processes (Hoffmann-Walbeck, 2022). Usually, the specific PDF file created with settings conforming to PDF/X is used for printing. Minimal individualization is possible for paper-based product catalogs using proven methods of print-on-demand, variable data printing, programmatic printing, and sub-catalogs.

Using print-on-demand, the catalog is produced on the basis of an interaction, e.g. an order. With variable data printing, single pages of the catalog are individualized, e.g. with an individual name (Lin, 2006). With programmatic printing, a combination of the before-mentioned possibilities is achieved, that has been possible since the advent of non-impact printing. For example, when an (online) shopping basket is not ordered, the customer receives a mailing with the products and a promotional code. Of course, programmatic printing is a major step forward in terms of multi- and omni-channel retailing, as now, preferences of digital channels can be used for the printing. However, the decisive factor is that the media that is produced shows a certain selection. This leads to a loss of information as a large number of products are not displayed. This is similar to the well-established paradigm of sub-catalogs (Angermann and Ramzan, 2016). Here, catalogs are printed that contain only products of certain categories (Xu, Du and Xu, 2014). For example, a sub-catalog is printed for a specific season, e.g. garden furniture for the summer. Sub-catalogs can also be used to filter the customers who receive the document (Mark, et al., 2019; de Melo, et al., 2019). For example, only customers who have bought garden furniture will receive the sub-catalog. However, dynamic preferences are not taken into account when producing the sub-catalogs (Angermann, 2022). It looks identical for all recipients, regardless of whether a customer lives in an apartment with a balcony, or in a house with a garden and pool. And maybe a homeowner is not only planning to buy new garden furniture, but also a garden kitchen. Unfortunately, kitchen appliances are not included in this sub-catalog, as those are logically part of an excluded product category. Like programmatic printing, this method results in a loss of information. And, in turn, both methods lead to a potential loss of sales. The homeowner may buy the kitchen appliances elsewhere or, in the worst case, the homeowner may look elsewhere to buy all the products at once. In summary, none of the existing methods provide a solution for paper-based product catalogs that takes into account individual preferences while still being able to include all products.

As a result, the only option for retailers that want to take customers preferences into account in the layout of the catalog is to layout the document manually using DTP software, or to accept a loss of potential sales.

To overcome above-mentioned limitations, the expert system (ES) TaxoCatalog is presented in this paper. The main difference to existing methods is that TaxoCatalog semantically personalizes the leading element for retailers regardless of branch or strategy, to publish content across channels. This is the product taxonomy, usually stored in a product information management (PIM) system. For portal-based media, the product taxonomy is used for product navigation on the e-commerce site. For paper-based media, the product taxonomy is used to structure documents for automatic layout setting of paper-based product catalogs using data-driven publishing. With TaxoCatalog, dynamic customer preferences can now be taken into account across channels based on this taxonomy, personalization is performed without loss of information, and automatic layout setting is still supported. To achieve this new approach to paper-based product catalogs, the proposed three-layered ES uses so-called background knowledge (BK).

The first layer of TaxoCatalog leverages two main sources of BK into its knowledge base (KB). These are the content-based BK of the taxonomy and possible semantic modifications, and the memory-based BK of customer preferences. Based on the first layer of BK, the second layer of TaxoCatalog infers semantically personalized taxonomies using a rule-based approach. For this purpose, different modification rules are provided by the included inference engine (IE). The third layer finally transforms the personalized taxonomy paths of the previous layer into the extensible markup language (XML) format.

The used architecture, especially because of the used BK, results in two other important advantages. Firstly, TaxoCatalog performs the computation of the semantically personalized taxonomy fully automatically, but the retailer is still able to customize the volume of personalization. And, the initial product taxonomy in the PIM remains consistent. In summary, the proposed ES TaxoCatalog provides three contributions to the data-driven publishing and printing industry:

- TaxoCatalog advances the interlinking of portal-based media, paper-based digital media, and printed media. This is achieved through the memory-based BK of customers' preferences.

- TaxoCatalog is the first solution for personalizing paper-based product catalogs without loss of information. This is achieved by using content-based BK of the taxonomy.

- TaxoCatalog is able to dynamically, yet fully automatically, scale the desired level of semantic personalization. This is achieved by using a rule-based IE.

The rest of the paper is organized as follows. Section Background describes the basic underlying methods necessary for the development of the proposed ES TaxoCatalog. After that, the ES is presented in detail in section TaxoCatalog. In section Case study, the ES is demonstrated in a real-world context. In section Evaluation, the ES is quantitatively and qualitatively evaluated from different perspectives. Finally, the conclusions are presented in the corresponding section.

2. Background

This section describes the methods necessary to develop the ES TaxoCatalog. Furthermore, this section will help to understand the use-case, methods, and implementation of the system presented in the following section. For this reason, four methods are necessary to be explained. These are the retail strategy paradigm called omni-channel retailing, the applications of taxonomy including its use in the context of omni-channel retailing, the principles of expert systems (ES) and the programming languages used, and the aim of data-driven publishing for the page-layout process.

2.1 Omni-channel retailing

The retail industry as a whole has been significantly impacted by digitization, particularly through the emergence of digital media channels (Hagberg, Sundstrom, and Egels-Zandén, 2016; Hübner, et al., 2021). This includes digital product distribution and sales channels (e.g. e-commerce, mobile commerce, external marketplaces), as well as digital marketing and hybrid marketing channels and techniques (e.g. social commerce, bots).

The demand to connect the heterogeneous channels into one single retail network represents the essential aspiration of today's retail industry (Hübner, Wollenburg and Holzapfel, 2016; Yrjölä, Saarijärvi and Nummela, 2018). The aim of networking the different channels is now to offer customers a so-called cross-channel customer experience (Asmare and Zewdie, 2022; Van Nguyen, McClelland and Thuan, 2022).

The most recent paradigm of maximum channel connectivity is called omni-channel retailing. This retail strategy is used by most of the leading retailers today (Jocevski, et al., 2019; Asmare and Zewdie, 2022; Li, et al., 2022). Compared to its predecessors, cross-channel and multi-channel, omni-channel retailing allows all channels that are part of the strategy to be supplied by a leading publishing source, and yet the channels can be interconnected without any barriers on the customer side (see Figure 1) (Verhoef, Kannan and Inman, 2015; Yrjölä, Saarijärvi and Nummela, 2018). In most cases, a PIM system is used as the leading source, structuring the product information using a product taxonomy (see section Taxonomy applications). Omni-channel retailing opens up new opportunities for both the retailer and the customer. First, the retailer can explore knowledge about the customer across channels in real time using recommender systems (RS). The RS typically combine statistical methods to measure the likelihood of ordering products. In particular, they use content and memory-based methods (e.g. order history, product features) or collaborative filtering (e.g. similar customers, similar ordering behavior). The results of recommender systems indicate customer preferences. These, in turn, can be used to display product-related content in a personalized manner on digital portal-based channels (Roy and Dutta, 2022). This opens up new opportunities for customers to receive personalized content across channels, creating a new kind of shopping experience throughout the customer journey (Alexander and Kent, 2022). For example, the customer receives a promotional code from an influencer on social media, sees the product on the main landing page when entering the online store, and purchases the product.

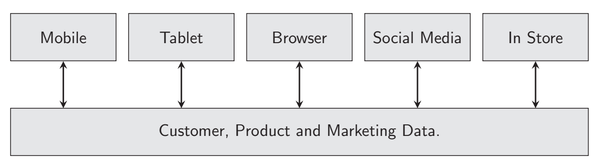

Figure 1: Omni-channel data integration schema (adapted from Cook, 2014)

2.2 Taxonomy applications

Taxonomies are a formal method of semantically structuring information using hierarchically ordered concepts (Angermann and Ramzan, 2017). The concepts can be described as a two-tuple: $\tau = \{\varphi, \rho\}$, where ρ is a set of edges for connecting partially ordered concepts of φ in taxonomy τ. Each edge represents a semantic hypernym–hyponym relationship between the concepts, expressed in simplified terms as a so-called *is-a* relationship. For example, 'Fish' *is-a* 'Seafood' (presented later in Figure 5).

Using the resulting string of *is-a* relationships, each concept can be uniquely defined by its so-called taxonomy path ρ_φ. Based on the unique paths for each concept, different concept types are distinguished. Concepts that share the same direct hypernym relationship are referred to as sibling concepts. For example, 'Fish' and 'Seaweed'. These concepts are also called sub concepts, as those specify the more general super concept. In our example, the super concept is 'Seafood'. Finally, the concept that is not further generalized is called the root concept, e.g. 'Products'.

Based on the number of φ that are part of ρ_φ, each path has a length, and each concept of the path has a depth. The deeper the position of one concept within the path, the more specific the concept is. The higher its position, the more general the concept is. Conversely, the path of one particular taxonomy with the maximum length defines the depth of the taxonomy. Formally, this is referred to as the number of levels N. In our example taxonomy, $N = 3$. Regardless of the depth and the underlying concept type, each concept is defined by a unique identifier (ID) and a label. This is usually a word or a multi-word in a natural language. Optionally, a concept can have a description, named gloss.

As taxonomies are able to semantically structure information based on the mentioned *is-a* principle, the represented information becomes knowledge (Bellinger, Castro and Mills, 2004). For this reason, taxonomies have many applications in information management (IM) (Angermann, 2017):

- In PIM, a taxonomy is used to hierarchically structure concepts in the form of product categories. This taxonomy is usually the leading element for all processes related to product information in the context of omni-channel retailing.
- In media asset management (MAM), the above-mentioned product categories are used to structure product-related media (e.g. images), and to assign those to the products.
- In e-commerce, content is usually provided by interfaces to other IM systems, e.g. PIM, MAM, enterprise resource planning (ERP), customer relationship management (CRM). Again, the product taxonomy is used as the leading element for data exchange.
- In data-driven publishing, the information from the PIM is used by the templates to automatically layout pages with dynamic content provided by the PIM and MAM.

As the examples above illustrate, the importance of taxonomy increases with the complexity of the IT landscape, which is the case with omni-channel retailing.

2.3 Expert systems

The ES are intelligent systems that aim to automatically infer decisions based on knowledge (Oleshchuk and Fensli, 2011). The main purpose of ES is to achieve a decision quality similar to that of a human expert (Mirmozaffari, 2019). Consequently, unlike machine learning (ML), the knowledge is not acquired by training an algorithm as for ML. However, ES have the ability to use, formally named consult, knowledge across expert systems. Thus, already defined knowledge can be consulted into the considered ES, as so-called BK (Angermann and Ramzan, 2017). These can be internally or externally generated BK from different sources.

In ES, the decisions are inferred using logic reasoning based on the mentioned knowledge, including BK (Merritt, 2012). This is another key difference from ML (Langley, 2011). The decision statements of ES are unambiguous (Ben-David and Frank, 2009). Each decision is true or false. A true statement states that the necessary knowledge is available to infer the decision, hence an unambiguous conclusion can be drawn. A false statement states that the necessary knowledge is not available to infer a decision. Before, all possible combinations of inferences are evaluated, called backtracking. Due to the binary answer spectrum, ES are well suited for taxonomic operations (Angermann and Ramzan, 2016). And, the ability of ES can be easily extended using BK (Angermann and Ramzan, 2017).

From a technical perspective, ES can be implemented using declarative programming languages (e.g. Python, Prolog). Prolog, in particular, is widely used for taxonomy-driven ES (Merrit, 2012). In Prolog, every program consists of two components (Bramer, 2005). These are a KB, and an IE. The KB represents knowledge in the form of fact predicates. The IE represents rule predicates to infer decisions based on the facts contained in the KB. Each rule uses facts, and/or other rules. Due to the interconnection between rules, and the backtracking, Prolog's inference mechanism can be time consuming. However, the computation can be accelerated using so-called dynamic facts. Then, the result of true statements is dynamically consulted into the KB as facts, formally called fact assert. The same is true for BK, as arbitrary sources can be easily consulted in the KB to further improve the Prolog program.

2.4 Page layout

The page-layout process is part of a print media production workflow (Angermann, 2023b). It aims to determine the layout of pages for paper-based documents, such as a printed product catalog. It involves the creative production of the design, depending on the desired format, the number of pages, and the graphical content to be included (Ambrose, Harris and Ball, 2019). Depending on the aforementioned criteria, this can be a time-consuming process when carried out manually using desktop publishing (DTP) software.

For this reason, different methods exist to automate the layout setting. These are methods known as scripting (e.g. JavaScript), or the database-driven methods (e.g. XML-based approaches).

Data-driven publishing, also known as database publishing, is a database-driven process that achieves the highest level of automation in the page-layout process. It is a rule-based technique for automatically setting the layout of paper-based documents using a layout generator. The software provides various data interfaces to databases of different formats, such as structured query language (SQL), comma separated values (CSV) and XML. In addition, the software provides a graphical user interface (GUI) for the designer to set up the data connections and to create the necessary components of the master page and templates (see Figure 2):

- The master page defines the dimensions of the paper-based document, and of the page frame(s). These frames are the areas of the paper that contain graphical content. Of course, each document can have multiple master pages.

- A template contains and defines so-called content frames for the page frames of master pages. The main purpose of the content frames is to include static and/or dynamic content, and to specify the design of this particular content. The main difference compared to other methods (DTP, scripting) is the use of dynamic content. This is content provided by a data source (e.g. a PIM). To do so, dynamic content is placed using data queries. For example, the product categories of products. A single content frame then consists of a rule that defines which content is to be dynamically queried and how this content should finally be designed. For example, the taxonomy path should be in 12 pt.

- A data source provides the dynamic content for each content frame as data stack. This stack is processed sequentially during the automatic generation of the document to be published. For example, a stack to query concepts as a sequence of paths.

The requirements for data-driven publishing logically lie in the data – more specifically, in the separation and structure of the data (Gündoğan, 2022). First, the data must be separated from channel-specific design, called media-neutral data storage. This allows the data to be later designed media-specific for different channels. Second, the data must be semi- or highly-structured. This results in the data marked using so-called tags. This means that the different data elements can be distinguished as distinct content types for data-driven publishing. For example, a content type 'concept' is used to filter out elements of the database (or data file) that are actually product categories. Based on this, all elements tagged as 'concept' are returned using the product path stack. Most omni-channel retailers fulfill data separation and data structure by using the hierarchical database format XML. This has the advantage that the data is semi-structured, and the data can be individually specified in the form of customizable tags (van der Vlist, 2002). In addition, the structure required for a particular document can be validated using the data format doctype definition (DTD), or other XML-based techniques (e.g. XML Schema).

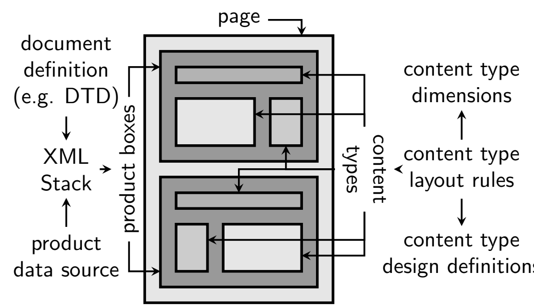

Figure 2: The concept and required components of data-driven publishing

In addition to data storage and structure, another important aspect of omni-channel retailing is data consistency. This means that all channels are based on the same (leading) data source. For example, the product information on the e-commerce site is based on the same data as the content displayed in the printed product catalog. In most cases, the PIM and product taxonomy is used as the leading element to publish consistent product information across online and offline channels (Angermann, 2022). In terms of product master data, the external procurement relationship (EPR) is usually the leading system.

3. TaxoCatalog

This section discusses the proposed ES TaxoCatalog for the semantic personalization of paper-based product catalogs in the context of omni-channel retailing. The comprehensive discussion starts by explaining the underlying use case, followed by the included methods, before explaining the implementation of the ES including its architecture using the logic programming language Prolog.

3.1 TaxoCatalog use-case

TaxoCatalog use-case is aimed at semantically personalizing product taxonomies for the layout setting of paper-based product catalogs based on customer preferences. These preferences are consolidated by a recommender system in an omni-channel context. This includes preferences derived from digital media, e.g. an e-commerce site, but it can also include preferences provided through other channels, e.g. a call-center, a local store or print media. For this reason, a current order does not necessarily have to be an authoritative interaction to infer a personalized product taxonomy using TaxoCatalog. Rather, the TaxoCatalog use-case is based on providing the customer, at a selected time *X*, with a semantically personalized printed product catalog *Y* (see Figure 3). The actors in the use-case involved are indirectly the customer, and the retailers' professional who decides to layout and print a personalized product catalog:

1. An initial product taxonomy provided by a PIM forms the basis of the use-case. In addition, a content-based BK detailing the semantic correspondences of the included concepts is entered.

2. The customer places orders via channels, e.g. on the e-commerce site. These individual preferences are entered into the ES as memory-based BK. Based on the BK, and the intention to layout a semantically personalized paper-based product catalog, the retailer's professional (e.g. designer, marketing expert), starts the computation, i.e. the rule-based IE. Now, the professional has to define the customer, and the volume of personalization desired. Logically, semantic correspondences, preferences, and the volume, are now taken into account to compute the customer-specific path, combined into a taxonomy. Here, different modification rules are used depending on the preferences.

3. The personalized taxonomy is exported by TaxoCatalog in XML file format. Logically, it contains paths with a semantic weight according to the customer's preferences.

4. The XML file is now used by the professional to start the layout setting processes using data-driven publishing software. Based on the semantic weight of the paths according to the customer's preferences, the designer can use standard queries that filter accordingly.

3.2 TaxoCatalog methods

TaxoCatalog uses different methods to infer semantically personalized product taxonomies. These are a content-based analysis method to derive semantic correspondences of the initial product taxonomy, a memory-based analysis method to result individual customer preferences, a modification rule-based method to personalize the taxonomy without loss of information, and finally a schema-based method to derive an XML file to be used for data-driven publishing.

3.2.1 Content-based analysis

The content-based analysis method uses natural language processing (NLP) to produce semantic correspondences of the original concepts in the form of mediator concepts. In TaxoCatalog, the output is based on the work presented in Angermann, Pervez and Ramzan (2017) and Angermann (2022). These are latent concept types of dependencies and collections (see Figure 4).

The concept type Dependency further specifies a super concept φ_I, and at the same time, further generalizes a set of sibling sub concepts φ_I_N specifying φ_I. Thus, a dependency forms a further, but latent level between super and sub concepts that are more synonymous than other sub concepts. For example, in the initial product taxonomy, the sibling sub concepts 'Soft Drinks', 'Coffees', 'Teas', 'Beers', and 'Ales' are assigned to the super concept 'Beverages' (see Figure 5). The dependency 'Alcoholic Drinks' further generalizes the sibling concepts 'Beers' and 'Ales' (see Figure 6). Same for the dependency 'Hot Drinks', which generalizes 'Coffees' and 'Teas', and for the dependency 'Nonalcoholic Drinks', which generalizes 'Soft Drinks'. The concept type Collection works in a similar way, but between the root concept and the super concepts. For example, the initial product taxonomy contains the super concepts 'Meat Poultry' and 'Seafood'. These

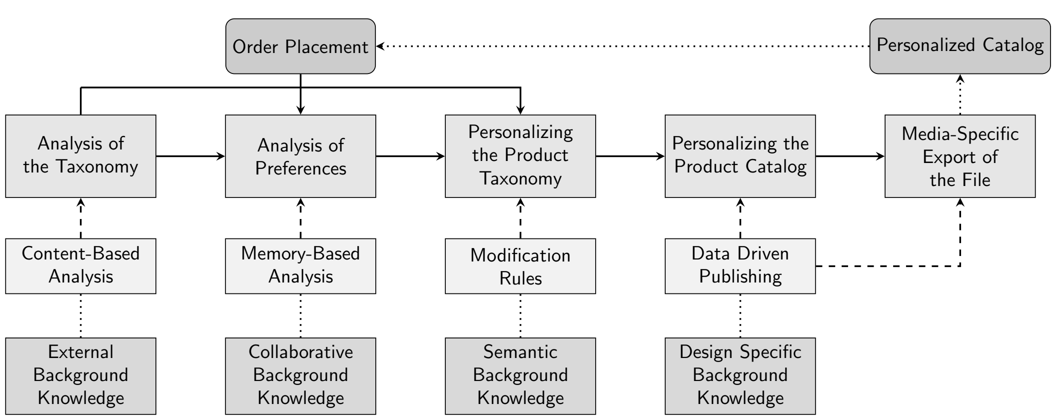

Figure 3: Use-case and methods of the ES TaxoCatalog

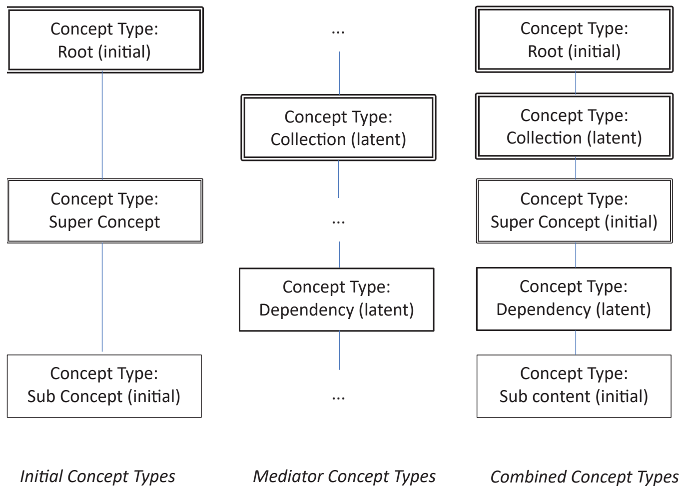

Figure 4: Hierarchical structure of a taxonomy using two mediator concept types

two super concepts can be generalized by a collection 'Meat and Seafood', etc. Both latent concept types, as well as the associated *is-a* relationships, are consulted in TaxoCatalog as external BK.

3.2.2 Memory-based analysis

The memory-based analysis method provides customer preferences using the recommender system as presented in the work of Angermann and Ramzan (2016). The method outputs preference statements. Each statement clarifies whether an individual customer has shown a low, medium, or high preference for similar product categories:

$$(\varphi\_I_{N1}, \varphi\_I_{NN}) \rightarrow \text{preference (low; medium; high)} \quad [1]$$

The algorithm first divides the orders into epochs, where more recent epochs are assigned with a higher epoch rate R. The algorithm then calculates the preferences for the different epochs E before combining them using $[R\_E]$. This is performed not only for one customer, but for customers sharing the identical customer group as well, known as collaborative filtering. This avoids over-specialization and the cold start problem (Desrosiers and Karypis, 2010). Again, the preference statements of the customers are consulted in TaxoCatalog as external BK. However, these preferences are further analyzed in TaxoCatalog using the above-mentioned method of latent concept types dependencies and collections. First, the preferences of the customers are consulted as a three-tuple:

$$\varepsilon_D = \{I, \rho_N, P\} \quad [2]$$

where I is the ID of the customer, ρ_N is a set of similar sub-concepts (dependency), and P is the preference statement (1 = low preference, 2 = medium preference, 3 = high preference). Second, the preferences of dependencies that generalize a collection are combined accordingly:

$$\varepsilon = \sum[\varepsilon_D]/N_{\varepsilon_D} \quad [3]$$

This results in a second and more general preference statement (1 = low preference, 2 = medium preference, 3 = high preference).

3.2.3 Modification rules

In TaxoCatalog, each taxonomy path is personalized according to the customer's preferences and the desired volume of personalization defined by the retailer's professional:

$$\text{Path Modification}_I = (P_C, P_D, M_C, M_D) \quad [4]$$

where P_C is the preference of a latent concept type collection for path I, M_C is the desired volume of personalization for the entire catalog, P_D is the perference of a latent concept type dependency for path I, and M_D is again the desired volume of personalization. Parameter M can have three states, corresponding to the different states of preferences; by default, $M = P$.

However, when the retailer decides not to modify the taxonomy according to the preferences, or not to modify a set of particular taxonomy path at all, the retailer can override P with M:

$$P \neq M \rightarrow P = M \quad [5]$$

Based on the Equation 5, it is clear that each path is modified separately, and in addition, both latent mediator concept types can now be modified separately. This is an improvement over recent work on taxonomy modification presented in Angermann (2022). For each latent mediator concept type, three modifications are available:

- $P = M = 1$ results in that the latent mediator concept is shown instead of showing the assigned more specific concepts. In terms of a collection, it means that the collection is shown instead of the super concepts. In terms of dependencies, it means that the dependency is shown instead of the sub concepts. In terms of $P_D = P_C = M_C = M_D = 1$, this means that the modified path considers the low preferences of the customer.

- $P = M = 2$ results in that the latent mediator concept is not shown, but the assigned more specific concepts are displayed. In terms of a collection, it means that the super concepts are shown instead of the collection. In terms of dependencies, it means that the sub concepts are shown instead of the dependency. In terms of $P_D = P_C = M_C = M_D = 2$, this means that the modified path considers the medium preferences of the customer.

- $P = M = 3$ results in that the latent mediator concept is shown, as well as the assigned more specific concepts. In terms of a collection, it means that the collection is shown, as well as the super concepts. In terms of dependencies, it means that the dependency is shown, as well as the sub concepts. In terms of $P_D = P_C = M_C = M_D = 3$, this means that the modified path considers the high preferences of the customers.

Of course, P_D and P_C of the individual paths can have different preference and volume statements (1, 2, 3). This results in that one single path can have nine possible modifications to be performed.

3.2.4 Schema export

TaxoCatalog is aimed to support automation of processes. For that reason, the personalized paths are transferred into an XML schema. This allows that the file can be processed using data-driven publishing software to automatically set the layout of the product catalogs to be printed. To transfer the single path into a valid well-formed XML, the following steps are performed:

1. The single concepts of one path are labelled with the tag `<concept>{Label}</concept>`. And, the ID of the concept is assigned to the start tag `<concept id="{ID Concept}">`.

2. The different concepts (3 to 5) of one path are assigned as sub-elements to the tag `<path>{Concepts}</path>`. The assignment is performed according to the depth of the concept within the single taxonomy path. For example, the root is the first element. In addition, the starting tag `<path id="{ID+ID+ID+(ID;{})+(ID;{})}">` is assigned an ID in the form of a concatenation of the IDs of the individual concepts.

3. All single paths of the personalized taxonomy are assigned as sub-elements to the tag `<customer>{Paths}</customer>`. Logically, the customer ID is assigned to the start tag `<customer id="Customer ID">`.

4. An XML header is assigned before the start tag to uniquely describe the customer. Finally, the exported XML looks like shown in Listing 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE customer SYSTEM "taxocatalog.dtd">
<customer id="1">
  <path id="1010000141001412014121">
     <concept id="10000">Products</concept>
     …
  </path>
</customer>
```

Listing 1: An extract from an XML file produced by TaxoCatalog

3.3 TaxoCatalog implementation

TaxoCatalog is implemented using the logic programming language Prolog. Logic programming languages are assigned to the declarative programming paradigm, which is often used for artificial intelligence applications. For that reason, a Prolog code does not serve as a collection of instructions, but as a description of the KB to infer logic conclusions (see section Expert Systems). For describing the KB, predicates expressed in the form of horn clauses are used. Each predicate has two components:

- The first component is the functor. It serves as the name of the predicate. For example, the predicate “concept(12000,”Trousers”).” has the functor “concept”.

- The second component are the arguments (1 to N), which stand in brackets behind the functor. The arguments are the knowledge assigned to the predicate. For example, the predicate “concept(12000,”Trousers”).” has two arguments, referred to as arity. The first argument is a number, the second argument is a string. Besides numbers and strings, arguments can also be compound terms or lists. A predicate ends by writing a dot “.” behind the enclosing bracket after the last argument.

Alternatively to the notation mentioned above, where the functor is mentioned with the arguments in detail, predicates can also be expressed by its short form. Then, the arity is written after the functor, separated by a slash, e.g. “concept/2”.

The predicate from the example given above is a so-called fact predicate. For example, the fact that a concept named ‘Trousers’ exists with the identifier of 12000, is expressed using “concept(12000,“Trousers”).”. Besides fact predicates, so-called rule predicates exist in Prolog. A rule predicate consists of two components to infer knowledge:

- The first component of a rule predicate is its head. It has a similar structure as a fact predicate: a functor, and arguments standing in brackets. For example, the rule predicate “twoconcepts(A,B) :- …” has a functor (“twoconcepts”) and two arguments (A,B). However, the arguments are usually not knowledge, but variables. In Prolog, variables are written with a capital initial letter.

- The second component is the body. Head and body are separated by a colon and hyphen “:-”. The body includes further rule and/or fact predicates, which are set in relation to each other using the logical *and* operator written as comma “,”, or by using the logical *or* operator written in the form of a semicolon “;”. For example, the rule predicate “twoconcepts(A,B) :- concept(A,_),concept(B,_).” includes two fact predicates of type “concept/2” to investigate if two different concepts exist. As can be seen, the variables from the head are used again in the body of the rule. By using variables, all possible combinations of relevant predicates are evaluated to test if a solution for the given problem exists inside the KB (true) or not (false). To ignore arguments not required for a specific rule, those can be ignored. This is done by using so-called anonym variables written as underscore “_”. Analogous to a fact predicate, a rule predicate ends by writing a dot “.”.

3.3.1 Program architecture

The ES TaxoCatalog is implemented using different layers, as suggested in recent literature discussing these types of intelligent systems (van der Aalst, Bichler and Heinzl, 2018). For TaxoCatalog, this means a three-layer approach. Doing so, only the data required for setting the layout is personalized. The initial data (in the leading PIM system) remains consistent:

- The storage layer consults all the required BK resources as fact predicates into the KB. This is the external BK based on the content-based analysis of the taxonomy and BK using the memory-based analysis of customer preferences. All resources of BK can be updated for each computation, if the underlying initial data changes in the leading systems. This is the product data in the PIM, but also the external BK mentioned before.

- The processing layer contains all the rule predicates within the IE necessary to compute a semantically personalized taxonomy for one customer. The main objective of the rules included is to create single and unique taxonomy path based on the customer’s preferences. Depending on the preferences, the semantics of the paths are adopted, resulting in combining concepts, splitting concepts, and changing levels.

- The publishing layer contains all the rule predicates within the IE of the ES that are necessary to export the semantically personalized catalog in a format that can be effectively input and processed by a layout generator software. For this reason, TaxoCatalog exports the semantically personalized taxonomy as an XML file. To do so, the different paths are transformed into XML tags, before the complete catalog is transformed into an enclosing custom XML tag, which is further enclosed by an enclosing customer XML tag.

As mentioned above, in Prolog the rules are related to each other. This is done using logical operators (*and* “,” , and *or* “;”). To optimize the processing time, dynamic facts are generated in TaxoCatalog. After the XML export has taken place, these dynamic facts are deleted (retracted) so that the program can be used for a new computation.

3.3.2 Program Sequence

The program sequence is structured according to the layers mentioned in the previous section. The top level rule predicate taxocatalog/0 contains the required rule predicates in its body (see Listing 2).

```
taxocatalog:-
 storagelayer,
 processinglayer,
 publishinglayer.
```

Listing 2: The upper-level rule predicate in TaxoCatalog

Within the rule predicate storagelayer/0 and its contained rule predicate readknowledge/0, as well as its contained rule predicates, e.g. concepts/1, different resources of the BK are consulted (see Listing 3). This is the external BK based on the content-based analysis of the taxonomy, and the BK based on the storage-based analysis of customer preferences (preferences/1).

In addition, the rule storagelayer/0 contains the rule predicate gettimestamp/0, which dynamically asserts the predicate timeidentifier/1 to the KB based on the current time and date. This is necessary to calculate different catalogs for different customers.

```
storagelayer:-
 gettimestamp,
 readknowledge.

readknowledge:-
 directory(Directory),
 string _ concat(Directory,'/Knowledge/',Folder),
 concepts(Folder),
 customers(Folder),
 dependencies(Folder),
 dependencymappings(Folder),
 collections(Folder),
 collectionmappings(Folder),
 preferences(Folder).

concepts(Folder):-
 string _ concat(Folder,"concepts.csv",File),
 csv _ read _ file(File,Rows,
  [functor(concept),arity(2)]),
 maplist(assert,Rows).
```

Listing 3: The rule predicates to consult used BK

After the rule storagelayer/0 is executed, the KB consists of seven dynamic facts describing knowledge and external BK. The fact concept/2 describes concepts of the initial product taxonomy using two arguments: ID and the concept's associated label. The customer/2 fact describes customers with an ID and a name. The fact dependency/2 describes dependency concepts with an ID and a label. The mapping of the different dependencies to the corresponding source concepts is done by the fact dependencymapping/3.

Each possible path extension is represented as a single fact using the ID of the super concept, the ID of the dependency and the ID of the sub concept (see Listing 4). Thus, the number of dependency mappings/3 is equal to the number of most specific concepts. Similarly to dependencies, two dynamic facts are used to represent the different collections: collection/2 and collectionmapping/3. Finally, the fact dependencypreferences/3 describes the individual customer preferences at the dependency level: Customer ID, Dependency ID, Preference.

Personalization of the catalog based on the dynamic BK fact predicates is performed within the rule predicate processinglayer/0, and the therein included rule predicates definecustomer/0, getpreferences/0, definevolume/0, and createcatalog/0 (see Listing 5). The main aim of the included rules is to use the different modification rules for single paths of the catalog depending on the customers individual preferences and the desired volume of personalization.

```
dependency(ID_Dependency,Label).

dependencymapping(ID_Super_Concept,
 ID_Dependency,ID_Sub-Concept).

collection(ID_Collection,Label).

collectionmapping(ID_Root_Concept,
 ID_Collection,ID_Super-Concept).

dependencypreference(ID_Customer,
 ID_Dependency,Preference_Degree).
```

Listing 4: The fact predicates for depicting mediator concepts and customers' preferences

```
processinglayer:
 definecustomer,
 getpreferences,
 definevolume,
 createcatalog.

getpreferences:-
 customer(ID_Customer),
 forall(collection(ID_Collection,_),
 collectionpreference(ID_Customer,
  ID_Collection)).

collectionpreference(ID_Customer,
 ID_Collection,Preference_Degree).

definevolume:-
 customer(ID_Customer),
  dependencyadaption(ID_Customer),
 collectionadaption(ID_Customer),
  setcustomerrules(ID_Customer).

createcatalog:-
 customer(ID_Customer),
 possiblepaths(ID_Customer),
 modification(ID_Customer),
 validatepaths(ID_Customer),
 removeduplicates(ID_Customer).
```

Listing 5: The rule predicates to infer semantically personalized Taxonomies

After the rule processinglayer/0 has been performed, the KB consists of several dynamic facts describing personalized paths of the taxonomy. First, the rule predicate definecustomer/0 asserts the current customer with an ID. Second, the rule predicate getpreferences/0, calculates and asserts the fact predicate collectionpreference/3, using dependencypreference/3.

Based on the two facts and the *is-a* relationships between the concepts of the initial taxonomy, it can be clearly stated which preference a customer has for one path of the initial taxonomy. Thirdly, the rule predicate definevolume/0 states and asserts the fact predicates dependencyadaptionrule/3 and collectionadaptionrule/3 to the KB.

```
path(ID_Customer,ID_Root,ID_Collection,
 ID_Dependency).

path(ID_Customer,ID_Root,ID_Collection,
 ID_Sub_Concept).

path(ID_Customer,ID_Root,ID_Super_Concept,
 ID_Dependency).

path(ID_Customer,ID_Root,ID_Super_Concept,
 ID_Sub_Concept).

path(ID_Customer,ID_Root,ID_Collection,
 ID_Dependency,ID_Sub_Concept).

path(ID_Customer,ID_Root,ID_Super_Concept,
 ID_Dependency,ID_Super_Concept).

path(ID_Customer,ID_Root,ID_Collection,
 ID_Super_Concept,ID_Dependency).

path(ID_Customer,ID_Root,ID_Collection,
 ID_Super_Concept,ID_Sub_Concept).

path(ID_Customer,ID_Root,ID_Collection,
 ID_Super_Concept,ID_Dependency,
 ID_Sub_Concept).
```

Listing 6: The fact predicates including personalized taxonomy paths

These rules define how a path should be treated based on the actual preferences and based on the desired degree of personalization. Consequently, for each type of mediator concept and for each type of possible preference degree, there is a fact with three arguments.

This is the degree of preference (1 to 3), the customer's ID and the desired modification. Based on this, two types of dynamic fact predicates are created to define the desired set, the dynamic fact predicates collectionrule/3 and dependencyrule/3. These define the actual desired modification for each mediator concept using the arguments customer ID, mediator concept ID and modification state. Thirdly, the rule predicate createcatalog/0 asserts the personalized path to the KB depending on the preferences and the desired scope of the modification. Paths are created for every possible and semantically correct combination. This results in three different dynamic fact predicates: path/4, path/5 and path/6. All three use the ID of the customer as the first argument and the ID of the root concept as the second argument. However, the other arguments, as well as the number of arguments, are different depending on the modification used (see Listing 6).

The XML export is achieved using the publishinglayer/0 rule predicate (see Listing 7) and the exportcatalog/0 rule predicate it contains. The retractcatalog/0 rule predicate removes dynamic facts.

```
publishinglayer:-
 exportcatalog,retractcatalog.

exportcatalog:-
 customer(ID_Customer),
 createpaths(ID_Customer),
 combinepaths(ID_Customer),
 writexml(ID_Customer),
 writefile(ID_Customer).
```

Listing 7: The fact predicates for exporting the personalized catalog in XML data format

Analogous to the other layers, dynamic fact predicates are asserted. First, the rule predicate createpaths/1 infers and asserts the fact predicate xmlpath/3 to the KB. Here, the first argument is the customer's ID, the second argument is the modification performed, and the third argument is paths in XML syntax. To do this, each concept of each path is assigned an associated XML tag. These elements are then collected and the collection is assigned an associated XML tag.

The result is a dynamic fact predicate called combinedpath/2, where the first argument is again the customer's ID, and the second argument is the XML element containing the path tag and the contained concept tags. Finally, the individual XML path elements are collected again and given an associated XML tag. This tag identifies the customer. This finally represents the XML schema of a personalized catalog.

4. Case study

As a demonstration of TaxoCatalog, let us consider the omni-channel retail market Northwind (2017), and three customers. The initial taxonomy consists of three levels and 31 concepts (see Figure 5).

Based on a content-based analysis of the taxonomy, the mediator concepts were consulted as BK, as well as the necessary mappings with the concepts of the initial taxonomy (see Figure 6). In detail, there are four collections and 16 dependencies. The taxonomy now has a maximum of five levels.

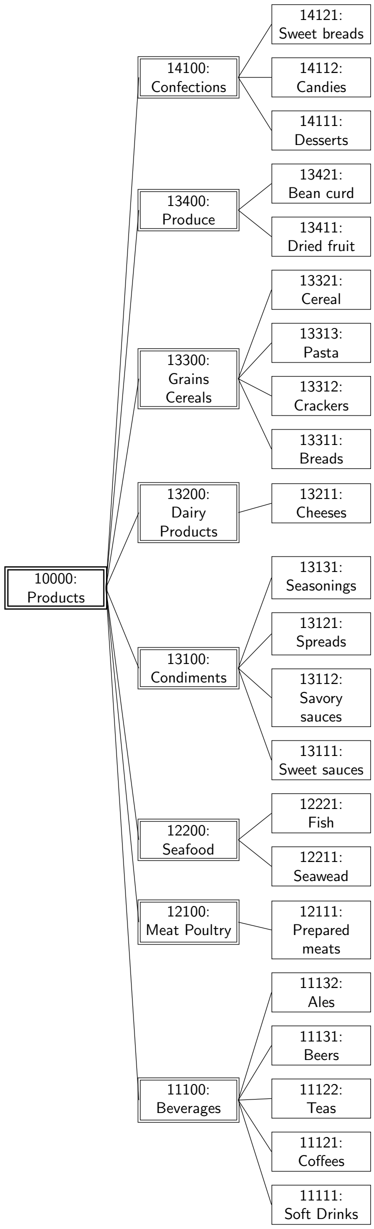

Figure 5: The initial product taxonomy of the omni-channel retailing market Northwind

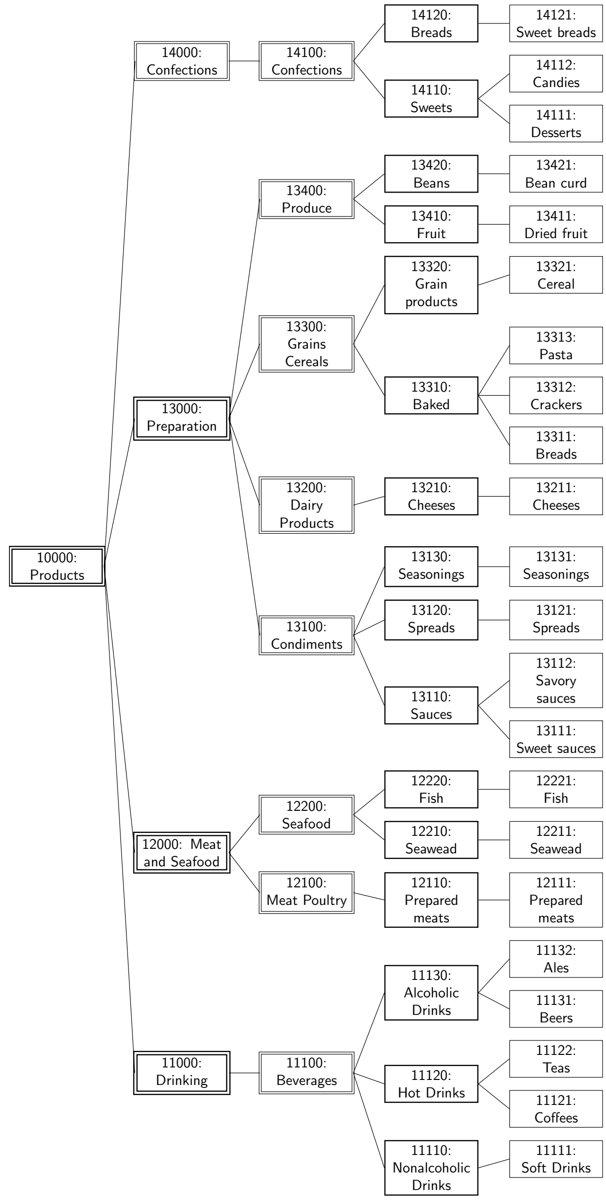

Figure 6: The initial product taxonomy of the omni-channel retailing market Northwind, including the two latent mediator concept types

4.1 Customers preferences

Before the personalization takes place, the preferences for the collections are inferred using TaxoCatalog's processing layer. This inference is based on the preferences shown for the dependencies. The resulting preferences for collections, are listed in Table 1.

In addition to the mediator concepts, the customer preferences are used in TaxoCatalog as BK. The customers considered (ID = 1, 2, 3) have shown the preferences listed in Table 2.

Table 1: Customers (ID = 1, 2, 3) inferred preference regarding collections (degree 1 = low, degree 2 = medium, degree 3 = high)

Collection (ID, Label)	**ID = 1 (degree)**	**ID = 2**	**ID = 3**
11000, Drinking	2	1	2
12000, Meat and seafood	2	2	3
13000, Preparation	2	3	2
14000, Confections	2	1	2

Table 2: Customers (ID = 1, 2, 3) consulted preferences regarding dependencies (degree 1 = low, degree 2 = medium, degree 3 = high)

Dependency (ID, Label)	**ID = 1 (degree)**	**ID = 2**	**ID = 3**
11110, Nonalcoholic Drinks	2	1	1
11120, Hot drinks	2	1	1
11130, Alcoholic drinks	2	1	3
12110, Prepared meats	1	2	3
12210, Seaweed	2	2	3
12220, Fish	1	2	3
13110, Sauces	1	3	1
13120, Spreads	2	3	1
13130, Seasonings	1	3	1
13210, Cheeses	1	3	1
13310, Baked	1	2	2
13320, Grain products	2	2	3
13410, Fruit	1	1	2
13420, Beans	1	1	1
14110, Sweets	1	1	2
14120, Breads	3	1	1

4.2 Catalog personalization

Each path is personalized according to the preferences for the dependencies, for the collections, and the desired volume of personalization. For the case study, it was assumed that $P = M$.

4.2.1 Customer 1

One customer (ID = 1) has medium preferences for all four collections, so the super concepts are used as in the original taxonomy (see Figure 7).

The customer has different preferences for the dependencies. For the six dependencies with medium preferences, the sub-concepts of the initial taxonomy are shown instead of the semantically more general dependency. For the nine dependencies with low preferences, the dependencies are shown instead of the individual sub-concepts. For the one dependency with high preferences, the dependency is shown as well as the individual sub-concepts, resulting in an additional level.

Based on this customer's preferences, a taxonomy of four levels and 28 concepts was created (Figure 7). Specifically, the semantically personalized taxonomy contains a root concept on level 1, eight super concepts on level 2, nine dependencies and eight sub concepts on level 2, and one sub concept on level 4. Compared to the initial taxonomy, the resulting personalized taxonomy consists of three fewer concepts and one additional level.

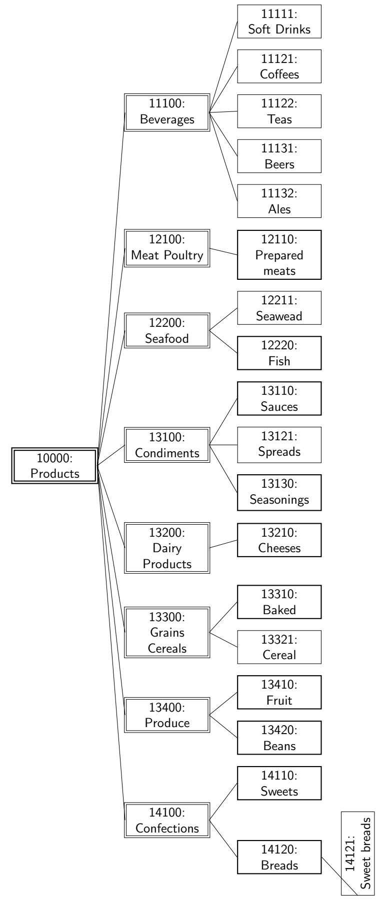

Figure 7: Semantically personalized taxonomy for Customer 1

4.2.2 Customer 2

Another customer (ID = 2) has different preferences for the four collections. For the two low preferred collections, those are shown instead of splitting those into semantically more general super concepts (see Figure 8). The opposite is inferred for the medium preferred collection. For the high preferred collection, the collection is shown, as well as the assigned super concepts. This adds a further level to the personalized taxonomy. For the dependencies, the customer has also heterogeneous preferences. Seven are low preferred; therefore here the dependencies are shown instead of the included sub concepts. Five are medium preferred, so here the sub concepts are shown without the dependencies. The remaining four dependencies are high preferred; this is the reason why those are shown and the sub concepts at a deeper level included. Now, the taxonomy has five levels.

Based on this customer's preferences, a taxonomy consisting of five levels and 33 concepts was created (Figure 8). In detail, the semantically personalized taxonomy includes one root concept on level 1, three collections and two super concepts on level 2, four super concepts, five dependencies and three sub concepts on level 3, six dependencies and four sub concepts on level 4, as well as finally, five sub concepts on level 5. Compared to the initial taxonomy, the resulting personalized taxonomy consists of two concepts more, and two additional levels.

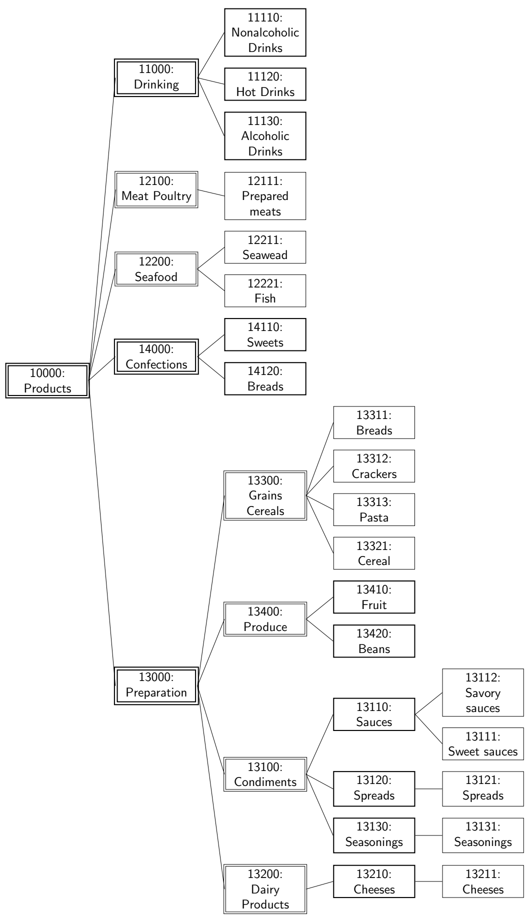

Figure 8: Semantically personalized taxonomy for Customer 2

4.2.3 Customer 3

The last customer (ID = 3) has different preferences for the four collections. For the high preferred collection, it is included in the paths, as well as the super concepts. This adds a further level to the personalized taxonomy (see Figure 9). For the three medium preferred collections, only the super concepts are shown. For the dependencies, the customer has also heterogeneous preferences. Eight are low preferred; therefore, here the dependencies are shown instead of the included semantically more specific sub concepts. Three are medium preferred, so the sub concepts are shown without the dependencies. Five, high preferred, are shown with the included sub concepts at a deeper level. Now, the taxonomy has five levels.

Based on this customer's preferences, a taxonomy consisting of five levels and 34 concepts was created (Figure 9). In detail, the semantically personalized taxonomy includes one root concept on level 1, one collection and six super concepts on level 2, two super concepts, ten dependencies and three sub concepts on level 3, three dependencies and five sub concepts on level 4, and finally, three sub concepts on level 5. Compared to the initial taxonomy, the resulting personalized taxonomy consists of three concepts more, and two additional levels.

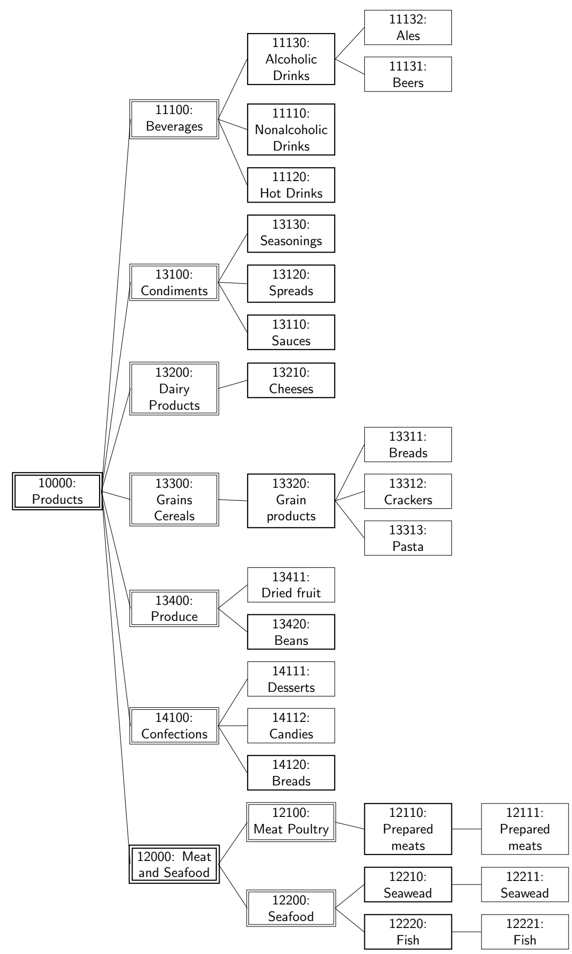

Figure 9: Semantically personalized taxonomy for Customer 3

4.3 Layout setting

The three personalized taxonomies are exported by TaxoCatalog as XML files. These files can be used for automatic layout generation using data-driven publishing software. For this project, the layout generator from the German software provider InBetween is used. All three files can be imported without any modifications compared to other XML files. The same template can be used for all customers, or an independent template can be created (see Figure 10). The template includes content frames with a query. In our example,

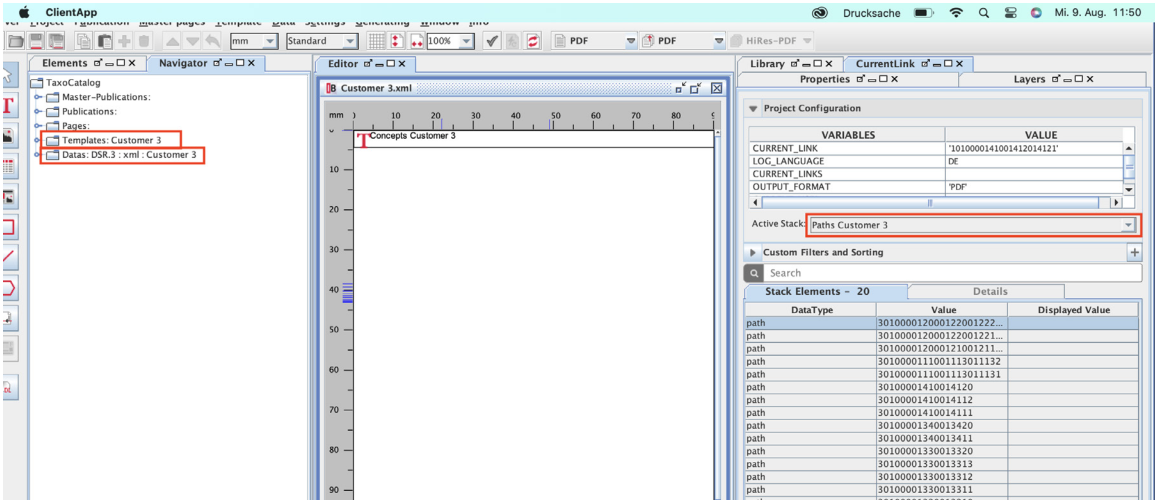

Figure 10: GUI of InBetween for layout setting using the XML output of TaxoCatalog

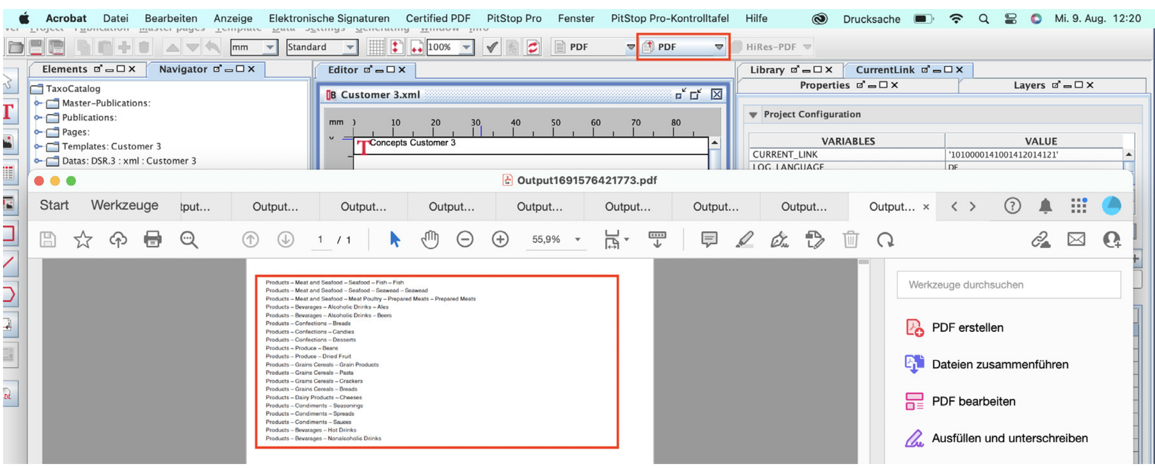

Figure 11: Dynamically generated publication using TaxoCatalog and InBetween

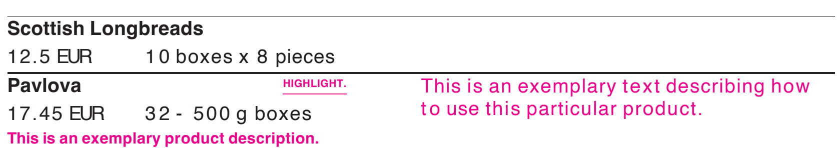

Figure 12: Personalization of product elements depending on semantic context weight

this means querying concepts sequentially using the path-driven stack. Based on the semantic weight of the path, the paths having a deeper level are at the top. Each stack element is queried, which results in the generation of a dynamic publication containing the paths (see Figure 11). Other templates making a separate use of the personalized semantic weights can also be created. This is of course the ultimate usage of TaxoCatalog XML output, but the focus of future research. For example, product information (e.g. gloss) can be included and designed based on the different semantic weights (see Figure 12).

5. Evaluation

The evaluation of rule-based ES is known to be challenging (Ligêza, 2006). This is because ES are designed to automate individual processes as performed by a human domain expert. Consequently, the quality of the ES is highly dependent on the expert. And, the implementation of the ES is usually done by using BK. Obviously, the quantity and quality of the BK used has a significant impact on the quality of the inference. To address the above aspects, the ES TaxoCatalog is evaluated in five directions:

- First, a comparison with previous studies for the research paradigm of taxonomy personalization including taxonomy modification is performed. This comparison examines the strengths and limitations of prior research efforts to give context of the proposed ES TaxoCatalog.

- Next, the BK efficiency is measured to check whether the BK used most effectively supports the methods contained in TaxoCatalog. As the BK is provided as output from other systems, a summary of the results is presented as a basis for evaluation.

- Based on above, the catalog modification efficiency of the ES TaxoCatalog is discussed to identify if the included modification rules effectively support personalizing a paper-based catalog.

- Following, the workflow integration efficiency of TaxoCatalog is discussed to determine if the ES can be integrated into existing print media workflows without changing those workflows.

- Finally, the catalog personalization efficiency of TaxoCatalog is discussed to verify how the semantically personalized taxonomies are recently used by the system and how the semantically personalized taxonomies could also be used in terms of layout setting. Consequently, this discussion will reveal open directions for future research of the proposed expert system.

5.1 Comparison with previous studies

The comparison with previous studies examines the strengths and limitations of prior research efforts to give context of TaxoCatalog. Consequently, previous studies are discussed, which focus on the aspect of taxonomy personalization and modification.

The very first study discussing the need for taxonomy modification was presented in Joh and Lee (2003). The core element of the Prolog-based implementation is different taxonomy modification rules that have to be performed manually by a user. A study discussing the need of intelligent-driven modification was presented in Lin and Hong (2008). Here, a database model collects knowledge about the customer and the taxonomy, to help the expert manually generating new taxonomies. Two similar frameworks, but with the capability to analyze about the customers and the taxonomy, were presented in Pierrakos and Paliouras (2010), and Tao, Li and Zhong (2011). In 2015, the need of preference-based taxonomy was additionally discussed in Svee and Zdravkovic (2015), before the first work allowing a fully-automatic modification and personalization was presented in Angermann and Ramzan (2016). Here, an ES implemented in Prolog was provided. This ES is able to analyze customers shopping behavior for personalizing the taxonomy based on predefined modification rules. In addition to earlier approaches, the ES is additionally capable to consider the semantics of the taxonomy. This means that the information remains semantically consistent after modifying the taxonomy, by using so-called mediator concepts instead of only limiting the modification to the initial concepts. Analogous to all other above-mentioned studies, their ES is aimed for e-catalogs, meaning the modification of taxonomies to be used for digital publishing channels, especially e-commerce. This is the same for the subsequent work presented in Mao, et. al (2020). Here, a framework was presented that can enrich the taxonomy based on self-supervision. And, it is the same for the work presented in Pawlowski (2021). Here, a ML approach was presented to analyze customers word usage for personalizing the labels of the concepts inside the initial taxonomy. The very last study, and the first study discussing the need of personalizing taxonomies for non-digital channels as well, was presented in Angermann (2022). This work includes a set of different modification rules to modify the taxono-

Table 3: Phases and works on taxonomy personalization and modification

Phase	Description	Manual/ Automatic	References
1	Taxonomy modification rules	Manual	(Joh and Lee, 2003) (Angermann, 2022)
2	Taxonomy personalization using taxonomy modification rules	Manual	(Lin and Hong, 2008) (Pierrakos and Paliouras, 2010) (Tao, Li and Zhong, 2011)
3	Taxonomy personalization using taxonomy modification rules inside intelligent expert systems	Automatic	(Angermann and Ramzan, 2016) (Mao, et al., 2020) (Pawlowski, 2021)
4	Taxonomy personalization for different channels using taxonomy modification rules	Automatic	(Angermann, 2023a) TaxoCatalog

mies based on the capabilities of the different channels. In contrast to other works, the rule-based approach includes various interchangeable modifications, as well as the capability of mediator concepts. However, it has to be performed manually.

Summarizing the different previous studies on taxonomy personalization and modification, four phases or research can be identified (Table 3). In the first phase, the focus was on providing taxonomy modification rules that have to be performed manually. In the second phase, the focus was still on modifying taxonomies manually, but based on knowledge about the customers. In the third phase, user behavior and modification rules were combined to allow a fully-automatic personalization using intelligent systems. TaxoCatalog can therefore be considered as the fourth phase building upon the three previous phases of research on taxonomy modification and personalization. It performs taxonomy modification rules fully automatically based on the capabilities of phase three, but is implemented to perform in a cross-media context.

5.2 Background knowledge efficiency

The BK efficiency used in TaxoCatalog was evaluated with regard to a wide range of criteria and databases (see Table 4). Results for the two most relevant metrics are summarized in this section. These are the *F*-measure of the memory-based BK, and the semantic flexibility efficiency of the content-based BK regarding semantic correspondences.

The *F*-measure score indicates the decision quality of a recommender system for analyzing customer preferences compared to a human domain expert. A score of 1 means that the correct decision was made in all test cases. A score of 0 means the opposite. The average *F*-measure score of the memory-based BK used in TaxoCatalog is 0.94 for all three databases (Angermann and Ramzan, 2016): 0.90 (Northwind), 0.93 (Adventureworks), and 0.98 (Festool). It is particularly noteworthy that the score is similar for all three databases: σ = 0.033. This means that an equally good quality result can be achieved for a variety of product areas, i.e. for omni-channel retailers in different sectors.

Semantic flexibility efficiency is measured to verify the flexibility of using a taxonomy as the leading element for personalizing a product catalog. Two scores are important here. First, the reduction flexibility is calculated by identifying the maximum possible reduction of the taxonomy. This is done for mediator concepts (collections, dependencies) that are less preferred by a particular customer. Secondly, the extension efficiency is calculated to verify a maximum possible reduction of the taxonomy. This is done for mediator concepts (collections, dependencies) that are highly preferred by the same particular customer. The average reduction flexibility score of the content-based BK used in TaxoCatalog is 48.10 %, and the extension flexibility score is 51.26 % (Angermann, Pervez and Ramzan, 2017).

5.3 Catalog modification efficiency

In TaxoCatalog, the BK mentioned above is used as input for the KB. However, this BK is further processed in a separate IE. Consequently, this IE must also be evaluated. The discussion of its catalog modification efficiency aims at finding out whether the contained modification rules also support the semantic personalization of a paper-based printed product catalog in the most effective way. The discussion presented in this section is therefore the logical application-oriented evaluation based on the qualitative results presented above (see section Comparison with previous studies).

In TaxoCatalog, three path modifications are provided based on the two underlying mediator concept types: dependencies, collections. As each initial taxonomy path always contains both latent mediator concepts, each path can be modified using nine different options. Each mediator concept can be displayed without including the initially assigned concepts, each mediator concept can be hidden and the initially assigned concepts displayed, and each mediator concept can be displayed including the assigned concepts at the deeper level. Consequently, each path can be extended by two levels and by the number of mediator concepts ($N = N + 2$). In this way, all paths that share an *is-a* relationship (sibling concepts) can be combined by their included dependency and collection.

Table 4: Characteristics and parameters of the databases used for experimental results

N	Adventureworks	Northwind	Festool
Customers	700	93	500
Orders	31 464	829	1 400
Products	320	77	118
Concepts (initial)	1 + 4 + 37 = 42	1 + 8 + 22 = 31	1 + 9 + 43 = 53
Dependencies	14	16	23
Collections	2	4	3

Based on the modifications, each individual path is semantically correct on its own due to the external BK used. However, as shown in the case study (see section Catalog personalization), different concept types become concepts of the same level through the implemented rules. This leads to conceptual heterogeneity, even if each path is semantically correct (Angermann, Pervez and Ramzan, 2017). This is the case when concepts with different semantic weights are displayed at the same level. Whether this is actually a challenge for the customer would need to be examined separately in a user study.

5.4 Workflow integration efficiency

The use of TaxoCatalog promotes the linking of portal-based media, paper-based digital media and print media. But it does mean that new software has to be integrated into an existing workflow.

The discussion of workflow integration aims to determine whether the ES can be integrated into existing workflows without having to adapt these workflows. This is important because workflows in the printing industry can be complex, depending on variables such as product variety, technical heterogeneity and the degree of automation of the processes involved (Hofmann-Walbeck, 2022). The following discussion assumes that catalogs are printed using non-impact printing.

Firstly, the TaxoCatalog ES can be implemented in existing workflows without the need to adapt processes. This is the case for upstream and downstream processes that are set up as state of the art for publishing content in an omni-channel context. There are two main reasons for this. Firstly, TaxoCatalog uses external BK and knowledge as provided by the latest omni-channel publishing IT infrastructures. This is media-neutral data provided by a PIM system. Secondly, TaxoCatalog transforms the paths into the standard file format required for data-driven publishing. This is achieved by transforming them into XML format.

However, the ES assumes that the given workflow is as shown above. In other words, the ES requires structured, media-neutral data as provided by a PIM. And it assumes that the subsequent page-layout process is actually performed using data-driven publishing. If one of these aspects (structured data, data-driven publishing) is not taken into account in the workflow, it will not be supported. For example, if DTP is used instead.

5.5 Catalog personalization efficiency

The aim of TaxoCatalog is to provide an ES for semantically personalizing paper-based product catalogs in omni-channel context using BK. Finally, its catalog personalization efficiency is discussed with respect to real retail scenarios.

Regardless of the media channel used (digital portal-based, digital paper-based, printed paper-based), in most cases the ultimate element for structuring the content to be included in a particular product catalog is the underlying product taxonomy. On an e-commerce site as well as for a printed catalog, the (identical) taxonomy is used as the main element for navigation and structuring. As a result, products are displayed in the corresponding catalog sections. This is because products are assigned to the concepts (product categories) of the taxonomy as so-called members – for example, on category-specific landing pages on an e-commerce site, or on category-specific pages in a printed product catalog. For this reason, the taxonomy is also the leading element when it comes to automatically setting the layout of paper-based product catalogs using data-driven publishing.

TaxoCatalog is used to perform semantic personalization, as mentioned above. This means that the initial product taxonomy is semantically personalized based on BK. This is knowledge about the customer and other knowledge about the initial taxonomy. The BK is used to combine, split and expand product categories according to the individual's preferences. As a result, the personalized catalog is structured according to these specific individual preferences. Thus, from a semantic point of view, even the most essential element of a catalog is personalized to the maximum extent that can be achieved with the help of TaxoCatalog. However, this personalization may not be enough for the customer. While the taxonomy is the essential element from a semantic point of view, the personalization of the content can be a compelling criterion for the customer in terms of personalization based on preferences.

In this regard, semantic context weighting will play a crucial role in subsequent research work. TaxoCatalog is not only designed to personalize taxonomies. It is also designed to personalize content using the personalized taxonomies. In concrete terms, this means that elements from a set of product information can be displayed in the catalog or not, depending on the semantic context weight. For example: a path that has been semantically enriched to the highest level will show products with all available information (e.g. title, description, price, sample application) and will also be highlighted by design. In contrast, a path that has been semantically reduced to the highest level shows only selected information (e.g. title, price). A corresponding example is already shown in this study (see section Layout setting). This potential will be explored in more detail in a later research project.

6. Conclusions

This work presented TaxoCatalog, an expert system for semantically personalizing paper-based product catalogs in an omni-channel retail context using background knowledge. Compared to the method of sub-catalogs, the presented expert system performs the personalization without any loss of information. This is achieved by using different sources of background knowledge and by applying different modification rules based on this background knowledge. The evaluation performed in different directions underlines the efficiency of the presented expert system – in terms of catalog modification efficiency, workflow integration efficiency and catalog personalization efficiency. Compared to previous studies in the field, TaxoCatalog works fully automatically, without suffering from information loss, and can be used across publishing channels.On the basis of the extensive evaluations and the included case study, two main directions for future research could have been identified. Firstly, the resulting conceptual heterogeneity can be investigated in a user study. Secondly, the personalization of product information based on the proposed system should be further investigated. Here, the resulting semantic context weight can be used as a main starting point for future research projects.

Acknowledgements

Thanks to the Festool GmbH & Co. KG and the InBetween Deutschland GmbH for providing data and software for the presented research.

References

Alexander, B. and Kent, A., 2022. Change in technology-enabled omnichannel customer experiences in-store. *Journal of Retailing and Consumer Services*, 65: 102338. https://doi.org/10.1016/j.jretconser.2020.102338.

Ambrose, G., Harris, P. and Ball, N., 2019. *The fundamentals of graphic design*. 2nd ed. London: Bloomsbury Publishing.

Angermann, H. and Ramzan, N., 2016. TaxoPublish: towards a solution to automatically personalize taxonomies in e-catalogs. *Expert Systems with Applications*, 66, pp. 76–94. https://doi.org/10.1016/j.eswa.2016.08.058.

Angermann, H., and Ramzan, N., 2017. *Taxonomy matching using background knowledge: linked data, semantic web and heterogeneous repositories*. Cham: Springer.

Angermann, H., Pervez, Z. and Ramzan, N., 2017. *Taxo-Semantics*: assessing similarity between multi-word expressions for extending e-catalogs. *Decision Support Systems*, 98, pp. 10–25. https://doi.org/10.1016/j.dss.2017.04.001.

Angermann, H., 2017. *Manager's guide to SharePoint server 2016: tutorials, solutions, and best practices*. New York: Apress Media/Springer.

Angermann, H. (2022). *TaxoMulti*: rule-based expert system to customize product taxonomies for multi-channel e-commerce. *SN Computer Science*, 3(2): 177. https://doi.org/10.1007/s42979-022-01070-8.

Angermann, H., 2023a. TaxoCatalog: costless workflow integration of an expert system environment to personalize product catalogs in omni-channel context. In: C. Ridgway, ed. *Advances in Printing and Media Technology: Proceedings of the 49th Research Conference of iarigai*. Wuppertal, Germany, 18–20 September 2023. Darmstadt: iarigai.

Angermann, H., 2023b. Interaction design for print media – Teaching print media workflows by emphasizing the great diversity of print media products. In: A. Politis, ed. *Proceedings of the 54th Scientific Conference of the International Circle of Graphic-Media Technology and Management*. Wuppertal, Germany, 18–20 September 2023. Athens, Greece: HELGRAMED.

Asmare, A. and Zewdie, S., 2022. Omnichannel retailing strategy: a systematic review. *The International Review of Retail, Distribution and Consumer Research*, 32(1), pp. 59–79. https://doi.org/10.1080/09593969.2021.2024447.

Beck, N. and Rygl, D., 2015. Categorization of multiple channel retailing in multi-, cross-, and omni-channel retailing for retailers and retailing. *Journal of Retailing and Consumer Services*, 27, pp. 170–178. https://doi.org/10.1016/j.jretconser.2015.08.001.

Behera, R.K., Gunasekaran, A., Gupta, S., Kamboj, S. and Bala, P.K., 2020. Personalized digital marketing recommender engine. *Journal of Retailing and Consumer Services*, 53: 101799. https://doi.org/10.1016/j.jretconser.2019.03.026.

Bellinger, G., Castro, D., and Mills, A., 2004. *Data, information, knowledge, and wisdom*. [online] Available at: <http://www.systems-thinking.org/dikw/dikw.htm> [Accessed July 2023].

Ben-David, A. and Frank, E., 2009. Accuracy of machine learning models versus "hand crafted" expert systems – a credit scoring case study. *Expert Systems with Applications*, 36(3), pp. 5264–5271. https://doi.org/10.1016/j.eswa.2008.06.071.

Bramer, M., 2005. *Logic programming with Prolog*. Berlin: Springer Science+Business Media.

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M. and He, X., 2023. Bias and debias in recommender system: a survey and future directions. *ACM Transactions on Information Systems*, 41(3): 67. https://doi.org/10.1145/3564284.

Cook, G., 2014. Customer experience in the omni-channel world and the challenges and opportunities this presents. *Journal of Direct, Data and Digital Marketing Practice*, 15(4), pp. 262–266. https://doi.org/10.1057/dddmp.2014.16.

de Melo, T., da Silva, A.S., de Moura, E.S. and Calado, P., 2019. *OpinionLink*: leveraging user opinions for product catalog enrichment. *Information Processing & Management*, 56(3), pp. 823–843. https://doi.org/10.1016/j.ipm.2019.01.004.

Desrosiers, C. and Karypis, G., 2010. A comprehensive survey of neighborhood-based recommendation methods. In: F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, eds. *Recommender systems handbook*. New York: Springer Science+Business Media, pp. 107–144.

Gao, L.(X.), Melero, I. and Sese, F.J., 2020. Multichannel integration along the customer journey: a systematic review and research agenda. *The Service Industries Journal*, 40(15–16), pp. 1087–1118. https://doi.org/10.1080/02642069.2019.1652600.

Gündoğan, E., 2022. *Robotic Process Automation im Desktop-Publishing: Eine Einführung in softwaregestützte Automatisierung von Artwork-Processen*. Wiesbaden: Springer.

Hagberg, J., Sundstrom, M. and Egels-Zandén, N., 2016. The digitalization of retailing: an exploratory framework. *International Journal of Retail & Distribution Management*, 44(7), pp. 694–712. https://doi.org/10.1108/IJRDM-09-2015-0140.

Hänninen, M., Kwan, S.K. and Mitronen, L., 2021. From the store to omnichannel retail: looking back over three decades of research. *The International Review of Retail, Distribution and Consumer Research*, 31(1), pp. 1–35. https://doi.org/10.1080/09593969.2020.1833961.

Herhausen, D., Kleinlercher, K., Verhoef, P.C., Emrich, O. and Rudolph, T., 2019. Loyalty formation for different customer journey segments. *Journal of Retailing*, 95(3), pp. 9–29. https://doi.org/10.1016/j.jretai.2019.05.001.

Hoffmann-Walbeck, T., 2022. *Workflow automation: basic concepts of workflow automation in the graphic industry*. Cham: Springer Nature.

Hübner, A., Wollenburg, J. and Holzapfel, A., 2016. Retail logistics in the transition from multi-channel to omni-channel. *International Journal of Physical Distribution & Logistics Management*, 46(6/7), pp. 562–583. https://doi.org/10.1108/IJPDLM-08-2015-0179.

Hübner, A., Amorim, P., Fransoo, J.C., Honhon, D., Kuhn, H., Martinez de Albeniz, V. and Robb, D., 2021. Digitalization and omnichannel retailing: innovative OR approaches for retail operations. *European Journal of Operational Research*, 294(3), pp. 817–819. https://doi.org/10.1016/j.ejor.2021.04.049.

Jocevski, M., Arvidsson, N., Miragliotta, G., Ghezzi, A. and Mangiaracina, R., 2019. Transitions towards omni-channel retailing strategies: a business model perspective. *International Journal of Retail & Distribution Management*, 47(2), pp. 78–93. https://doi.org/10.1108/IJRDM-08-2018-0176.

Joh, Y.H. and Lee, J.K., 2003. Buyer's customized directory management over sellers' e-catalogs: logic programming approach. *Decision Support Systems*, 34(2), pp. 197–212. https://doi.org/10.1016/S0167-9236(02)00081-7.

Langley, P., 2011. The changing science of machine learning. *Machine Learning*, 82(3), pp. 275–279. https://doi.org/10.1007/s10994-011-5242-y.

Lin, C. and Hong, C., 2008. Using customer knowledge in designing electronic catalog. *Expert Systems with Applications*, 34(1), pp. 119–127. https://doi.org/10.1016/j.eswa.2006.08.028.

Lin, X., 2006. Active layout engine: algorithms and applications in variable data printing. *Computer-Aided Design*, 38(5), pp. 444–456. https://doi.org/10.1016/j.cad.2005.11.006.

Li, Z., Wang, D., Yang, W. and Jin, H. S., 2022. Price, online coupon, and store service effort decisions under different omnichannel retailing models. *Journal of Retailing and Consumer Services*, 64: 102787. https://doi.org/10.1016/j.jretconser.2021.102787.

Ligêza, A., 2006. *Logical foundations for rule-based systems*. 2nd ed. Heidelberg: Springer.

Mao, Y., Zhao, T., Kan, A., Zhang, C., Dong, X.L., Faloutsos, C. and Han, J., 2020. Octet: online catalog taxonomy enrichment with self-supervision. In: *KDD '20 Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Virtual event, CA, USA, 6–10 July, 2020. New York: ACM, pp. 2247–2257. https://doi.org/10.1145/3394486.3403274.

Mark, T., Bulla, J., Niraj, R., Bulla, I. and Schwarzwäller, W., 2019. Catalogue as a tool for reinforcing habits: empirical evidence from a multichannel retailer. *International Journal of Research in Marketing*, 36(4), pp. 528–541. https://doi.org/10.1016/j.ijresmar.2019.01.009.

Merritt, D., 2012. *Building expert systems in Prolog*. New York: Springer Science+Business Media.

Mirmozaffari, M., 2019. Developing an expert system for diagnosing liver diseases. *European Journal of Engineering and Technology Research*, 4(3), pp. 1–5. https://doi.org/10.24018/ejeng.2019.4.3.1168.

NorthWind, 2017. *NorthWind*. [online] Available at: <https://github.com/cjlee/northwind> [Accessed July 2023].

Norvell, T., Kumar, P. and Contractor, S., 2018. Assessing the customer-based impact of up-selling versus down-selling. *Cornell Hospitality Quarterly*, 59(3), pp. 215–227. https://doi.org/10.1177/1938965518762836.

Oleshchuk, V. and Fensli, R., 2011. Remote patient monitoring within a future 5G infrastructure. *Wireless Personal Communications*, 57(3), pp. 431–439. https://doi.org/10.1007/s11277-010-0078-5.

Pawlowski, P., 2021. Non-deterministic logic of informal provability has no finite characterization. *Journal of Logic Language and Information*, 30(4), pp. 805–817. https://doi.org/10.1007/s10849-021-09344-9.

Pierrakos, D. and Paliouras, G., 2009. Personalizing web directories with the aid of web usage data. *IEEE Transactions on Knowledge and Data Engineering*, 22(9), pp. 1331–1344. https://doi.org/10.1109/TKDE.2009.173.

Roy, D. and Dutta, M., 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1): 59. https://doi.org/10.1186/s40537-022-00592-5.

Svee, E.-O. and Zdravkovic, J., 2015. Towards a consumer preference-based taxonomy for information systems development. In: R. Matulevičius and M. Dumas, eds. *Perspectives in Business Informatics Research: 14th International Conference, BIR 2015, Proceedings.* Tartu, Estonia, 26–28 August, 2015. Cham: Springer International Publishing, pp. 213–227. https://doi.org/10.1007/978-3-319-21915-8_14.

Tao, X., Li, Y. and Zhong, N., 2011. A personalized ontology model for web information gathering. *IEEE Transactions on Knowledge and Data Engineering*, 23(4), pp. 496–511. https://doi.org/10.1109/TKDE.2010.145.

van der Aalst, W.M.P., Bichler, M. and Heinzl, A., 2018. Robotic process automation. *Business & Information System Engineering*, 60(4), pp. 269–272. https://doi.org/10.1007/s12599-018-0542-4.

van der Vlist, E., 2002. *XML schema: the W3C's object-oriented descriptions for XML*. Sebastopol, CA: O'Reilly Media.

Van Nguyen, A.T., McClelland, R. and Thuan, N.H., 2022. Exploring customer experience during channel switching in omnichannel retailing context: a qualitative assessment. *Journal of Retailing and Consumer Services*, 64: 102803. https://doi.org/10.1016/j.jretconser.2021.102803.

Verhoef, P.C., Kannan, P.K. and Inman, J.J., 2015. From multi-channel retailing to omni-channel retailing: introduction to the special issue on multi-channel retailing. *Journal of Retailing*, 91(2), pp. 174–181. https://doi.org/10.1016/j.jretai.2015.02.005.

Xu, Z., Du, D. and Xu, D., 2014. Improved approximation algorithms for the max-bisection and the disjoint 2-catalog segmentation problems. *Journal of Combinatorial Optimization*, 27(2), pp. 315–327. https://doi.org/10.1007/s10878-012-9526-3.

Yrjölä, M., Saarijärvi, H. and Nummela, H., 2018. The value propositions of multi-, cross-, and omni-channel retailing. *International Journal of Retail & Distribution Management*, 46(11–12), pp. 1133–1152. https://doi.org/10.1108/IJRDM-08-2017-0167.