Information Systems 91 (2020) 101469

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Design principles for the General Data Protection Regulation (GDPR): A formal concept analysis and its evaluation

Damian A. Tamburri

Technical University of Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 16 September 2019
Received in revised form 12 November 2019
Accepted 15 November 2019
Available online 20 November 2019
Recommended by Dennis Shasha

Keywords:
Privacy-by-design
GDPR
Formal-concept analysis

ABSTRACT

Data and software are nowadays one and the same: for this very reason, the European Union (EU) and other governments introduce frameworks for data protection — a key example being the General Data Protection Regulation (GDPR). However, GDPR compliance is not straightforward: its text is not written by software or information engineers but rather, by lawyers and policy-makers. As a design aid to information engineers aiming for GDPR compliance, as well as an aid to software users' understanding of the regulation, this article offers a systematic synthesis and discussion of it, distilled by the mathematical analysis method known as Formal Concept Analysis (FCA). By its principles, GDPR is synthesised as a *concept lattice*, that is, a formal summary of the regulation, featuring 144372 records — its uses are manifold. For example, the lattice captures so-called *attribute implications*, the implicit logical relations across the regulation, and their intensity. These results can be used as drivers during systems and services (re-)design, development, operation, or information systems' refactoring towards more GDPR consistency.

1. Introduction

In the era of Big Data, the connubial nature between data, software, and its engineering becomes so critical that the entire societal structure needs to discipline it by means of regulations aimed at, among others, reducing or avoiding the risks connected to wrongful processing of data itself [1] — a key example of such regulations is the European Union (EU) General Data Protection Regulation (GDPR).[1] GDPR is the prime example of a discipline to regulate data usage and protection across a sovereign state federation such as the European Union, and could have a massive influence over the usage of any software-intensive technology around the EU itself, regardless of its user, or purpose of use. At the time of writing, the regimented use of GDPR as a rule of law has been implemented for little over 90 days,[2] with uncompliance sanctions ranging up to 4% of the global annual company turnover or 20 Mil. EUR.[3] On one hand, industries all over the world are more and more in a GDPR 'fear frenzy'[4], afraid of non-compliance of their software designs, and with merely 24 months to prove themselves compliant.[5] On the other hand, research around the regulation is at an early stage, concentrating mostly anecdotally [2,3] on understanding the dimensions of complexity behind the regulation [4]. Little work has been done in the direction of formalising the GDPR (or similar regulations, for that matter) to the purpose of aiding GDPR-compliance *by-design*, meaning, to the purpose of supporting software development via GDPR-compliant requirements engineering, design or refactoring patterns, or Operations guidelines for GDPR [5,6].

To address the above gap, this manuscript reports a synthesis and software design principles [7,8] for the GDPR elaborated via a systematic synthesis and analysis of the regulation, obtained by applying a formal analysis method known as Formal Concept Analysis [9,10]. FCA is a disciplined method of deriving a concept hierarchy from a collection of objects and their properties expressed originally in structured text (e.g., a policy, or a snippet of source code). Each concept in the hierarchy represents the objects sharing some set of properties while each sub-concept in the hierarchy represents a subset of the objects (as well as a superset of the properties) in the concepts above it. Hierarchies recovered through FCA draw from, and are formalised into, the mathematical theory of *lattices* developed by Garrett Birkhoff [11], and reflect, paraphrasing from the original text, partially ordered sets in which "[...] every two elements have a unique supremum [a

E-mail address: dtamburri@acm.org.

1 http://www.privacy-regulation.eu/en/.

2 https://howmanydaystill.com/its/gdpr.

3 https://www.taylorwessing.com/globaldatahub/article-enforcement-sanctions-under-gdpr.html.

4 https://www.linkedin.com/pulse/gdpr-nothing-fear-josh-mayfield/.

5 https://www.theregister.co.uk/2017/04/07/gdpr_people_not_process_your_worst_nightmare/.

https://doi.org/10.1016/j.is.2019.101469

least upper bound equivalent to a join in SQL syntax] and a unique infimum [also called a greatest lower bound]".

The use of FCA is costly [12], but justified by the formality and rigour with which documents and policies such as the GDPR are structured with. FCA is designed to recover the concepts, attributes, and implications hidden in a regulation, with the same formality and rigour with which the regulation was composed upon its creation [12]. By means of the method, in the context of GDPR a total of 144372 records were recovered and captured in an FCA matrix for further analysis.

The FCA analysis of GDPR revealed 4 practical design principles to be taken by software for GDPR compliance, namely: (1) *re-design for data protection officers* — appoint a data protection officer and restructure software designs in line with their technical needs and organisational demands; (2) *design for data-protection measures and metrics* — envision technical and organisational measurement that enables monitoring and control of data processing steps explicitly as well as identifying/profiling all data processors, regardless of their technical, organisational, or human nature; (3) *design for multiple data-levels* — distinguish between data processing for children and data processing for adults, allowing for appropriate rules and regulations for each, based on the appointed responsible party in either case; (4) *design for code-of-conduct* — devise organisational codes-of-conduct to increase awareness, training, and consistency of GDPR-compliance under the responsibility of the data protection officer and re-design software solutions to support such codes. Finally, to assess the veracity of revealed findings, two industrial case-studies were set-up according to the guidelines of Yin et al. [13] and involving a non-governmental organisation (NGO) as well as a municipality active in The Netherlands. The case-studies aimed at mapping the FCA-based insights and conclusions with (1) roles whose software support is essential for GDPR compliance stemming from the case-studies, (2) design challenges which are critical to be addressed for such compliance in the scope of the studied cases, as well as (3) any specific design constructs requiring special attention from software and information systems architects and observable in the cases in object. The conclusion of this evaluation is that indeed the FCA insights provide a lens to start from while designing for GDPR.

Beyond the practical usages implied by the above findings, results of FCA analysis of the regulation reveals that: (1) GDPR already provisions for GDPR-compliance *by design* — more specifically, the "compliance" concept identified across the GDPR links to the provisions that the regulation contains to aid compliance; (2) the complexity of GDPR can be allocated to the responsibility of a relatively simple organisational structure already embedded into the regulation — a first step in this direction is the immediate appointment of a Data-protection Officer in organisations as well as design software solutions to support the officer's role and demands (e.g., raise awareness, monitor compliance, train for compliant software design); (3) the regulation is designed for data defence, but it is far from being only restrictive, rather, it provides formalised guidelines for compliant software along two equally important dimensions, technical, and organisational.

The practical benefits of the FCA matrix, the lattices that were identified as well as the considerations recapped above are manifold, among which: (a) designers can use the recovered concept lattices, as a basis for GDPR-compliance *by design* — software designers require to consider every concept, relation, and attribute in each recovered lattice as a design driver [14] for their software architecture preparation, evaluation, and refactoring; (b) software operators can use auto-generated attribute implications as properties to be verified (e.g., with testing [15] or formal verification [16]) and metrics for the operational monitoring of software applications during their operation and continuous evolution [17]; (c) managers or data-officers in companies can use the synthesis provided in this manuscript as a starting point to evaluate and adapt their respective software and organisations in compliance with the regulation; (d) policy-makers can use the synthesis offered in this manuscript to increase its readability and accessibility to software professionals, thus encouraging compliance; (e) GDPR researchers can use the results to support further studies around the regulation by operational or semi-automated means; (f) security and privacy companies can use the synthesis to design compliance tools as well as further methodological support more in line with GDPR.

In conclusion, the exploration formalised in this manuscript can be used as a starting point for further research and practice along the same lines; furthermore, the FCA representation itself can be used as a staging tool to understand and operationalise GDPR compliance.

Finally, for the benefit of further research and practice along these lines, a replication package is provided for all practitioners and researchers to freely explore the results used to prepare this manuscript.[6]

Structure of the Paper. The rest of the paper is structured as follows. First, Section 2 offers background, including terms and definitions around GDPR and then moves to outline briefly the state of the art around the regulation. Next, Section 3 discusses the chosen research method, how it was applied, and how the results reliability were evaluated. Further on, Section 4 describes our results, while Sections 5 and 6 evaluate and discuss the results, offering tentative usages from three perspectives: (a) software practitioner; (b) software user; (c) software business. Finally, Section 7 concludes the paper outlining venues for future work.

2. Background and related work

2.1. GDPR Context and intended use

The GDPR is the EU response to the proliferation of processing, unlawful or otherwise, of data available from online or offline sources and obtained by conventional or unconventional means. The processing in question is, by recognition of the GDPR itself, aimed at a specific business intent or purpose and ascribed to one or more enterprises and, as such, shall be regimented to establish and certify that the purpose in question does not violate the basic principles upon which the EU is founded and that citizens across the EU expect to see upheld. Up to early 2018, companies based across EU were required to uphold a regulation known as Directive 95/46/EC[7] which, although specific for individuals and their data, did not cover several processing dimensions (e.g., compliance, processing rights, withdrawal of consent) or degrees of processing automation which are currently procuring risks to people and assets across the EU.

The GDPR was designed to amend and repeal Directive 95/46/EC so that individuals and organisations across the EU are warranted the proper data protection.

2.2. GDPR: Terms and definitions

Paraphrasing from Wikipedia, Privacy by design is an approach to systems engineering initially developed by Ann Cavoukian and later formalised in a joint report on Privacy-Enhancing Technologies (PETs) [18]. The report was edited by a joint team of the Information and Privacy Commissioner of Ontario (Canada), the

6 https://tinyurl.com/y7myobke.

7 http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.

Dutch Data Protection Authority and the Netherlands Organisation for Applied Scientific Research in 1995. As part of the report, the following terms and definitions apply to the present study. First, a *regulation* in this context is intended as a set of *rules* to restrict, constrain, or otherwise regiment the *processing* of *data* belonging to one or more *persons*, intended as natural individuals with a recognisable biological derivation. With *processing*, the GDPR identifies a series of activities or analysis operations applied to a series of datums (i.e., the *data*) for the purposes of determining new knowledge or insights for a specific *purpose*. Finally, a *purpose* is intended as an expression of interest by an enterprise of some sort and size, over specific data that can be inferred through analysis, electronic or otherwise. Furthermore, the GDPR distinguishes the processing of data by automated means, e.g., wherefore automated decision-making applies.

2.3. The regulation in a nutshell

GDPR is fleshed-out in a document counting 261 pages.[8] arranged in 99 articles, organised in 11 chapters. The chapters individually address the context, roles, restrictions of all concepts and attributes in the regulation. More in particular, chapters 1 to 11 address: (1) general provisions, terms, and definitions; (2) principles; (3) rights of the data subject; (4) controller and processor restrictions, and obligations; (5) data-transfer restrictions, rules, and dynamics; (6) supervision and supervisory authority; (7) restrictions over the cooperation between multiple organisations and partnerships; (8) penalties and liability definitions; (9) specific processing scenarios and connected restrictions, penalties; (10) implementing acts for the European Economic Area (EEA); (11) final provisions and establishment of repealed directives, previous agreements and related reports. InterSoft consulting offers a browsable outline of the regulation with direct access to recitals (i.e., the regulation individual articles themselves)[9]

At a glance, in terms of which companies/organisations may be subject directly to the regulation, two general insights apply:

- The organisation in question processes personal data belonging to a specific individual (a "data subject") and is legally based in the EU, regardless of where the actual data processing takes place;
- The organisation is established outside the EU but markets, offers, provides, or otherwise engages EU citizens or organisations with goods or services to, or monitors the behaviour of, individuals within the EU;

The rest of the insights contained in the following pages allow companies and organisations subject to the regulation to garner basic, mathematically-proven insights on how to re-design their own software facilities for processing data in a manner which is more consistent with the regulation.

2.4. GDPR Research

The state of the art in GDPR research is still scarce and rather preliminary. For example, in Basin et al.[10] seminal work on the matter, the researchers focus on understanding, abstracting, and tentatively formalising brief extracts of the regulation for the benefit of policy analysis. Furthermore, related work on the matter has covered and researched ways in which to phrase privacy and cybersecurity policies in a fashion consistent with GDPR, e.g., by Diamantopoulou et al. [19] or Zerlang et al. [2]. From another lens of analysis, GDPR was analysed from a business perspective, in an effort to understand the regulation impact over revenue and marketability of data in EU and worldwide [3].

Overall, a systematic investigation of the regulation for the benefit of practitioners and users influenced by GDPR is still unavailable in the state of the art. Therefore, the benefits and contributions of this manuscript along the lines of GDPR-compliant software design, development, and operations are considerable.

2.5. Privacy-by-design

There has been a considerable quantity of research in the past on how to specify and enforce privacy requirements in the scope of *privacy-by-design* according to the European Networks and Information Security Agency (ENISA).[11] For example, Contextual Integrity (CI) [20] is a normative framework, that actually inspired this research, for modelling the flow of information between agents and reasoning on communication patterns that cause privacy violations. Barth et al. provide a formalisation of CI using a Linear Temporal Logic (LTL), with the goal of precisely specifying privacy laws [21]. The framework in question includes privacy relevant concepts like agent role, communication purpose, obligation and focuses on the type of information transmitted.

Similarly, Ni et al. define a Privacy-Aware Role-Based Access Control (P-RBAC) model, which essentially considers various privacy-relevant concepts (purpose, role, obligation) to specify access control policies [22]. P-RBAC does not target the case of streaming data. Moreover, the model does not forsee any mechanism for controlling the informativeness of accessed data. Carminati et al. propose an access control model for data streams, whose key feature is the ability to model temporal access constraints, like the time window during which access is allowed [23]. Access control policies are then implemented through a query re-writing algorithm that exploits a set of secure operators. Nevertheless, there is no explicit focus on privacy aspects.

Within the range and scale of processing intended in GDPR, Vigiles [24] offers an approach for implementing access control on MapReduce systems, which is implemented in terms of a middleware that applies fine-grained access control rules on the input data of a MapReduce job. Also in this case there is no explicit focus on privacy concepts, nor on streaming data where the interest is on temporal aspects.

Although several works for privacy protection as well as access control exist, there is no systematic synthesis of such works aimed at offering or increasing GDPR-compliance; the contributions provided in this manuscript are at the heart of providing such synthesis as a first step towards achieving GDPR-compliance *by-design*.

2.6. Requirements engineering for privacy-by-design

Several works have investigated the design and implementation of private-by-design systems, stemming primarily from a requirements engineering perspective, where privacy becomes a problem of *compliance*, i.e., strict and formally-verified adherence to bylaws and regulations [25]. Already in the early 2000's, researchers from that community have investigated ways to elicit requirements from legal texts, for example, Maxwell et al. [26] propose production rule models which are consistent to legal texts (such as GDPR) and useable in the scope of privacy-by-design. The proposed methodology is indeed well-fit to instrument formal verification of private-by-design systems but is limited to 4 sections of the HIPAA privacy rule enacted in the US. Similarly to these works, other perspectives consider the

8 Available online: http://www.privacy-regulation.eu/en/.
9 https://gdpr-info.eu/recitals/.
10 https://www.inf.ethz.ch/personal/basin/pubs/fc18.pdf.
11 https://www.enisa.europa.eu/about-enisa.

ways to strike an optimal trade-off between effective software functionality and processing privacy [27]; the general conclusions of these works is that privacy-by-design always comes at an expense both in terms of systems design and in terms of reduced overall systems functionality. In that respect, the work in this article offers key insights into the GDPR that may reduce such expense, especially in the early-stage requirements and design phases as well as the refactoring stage, e.g., when a previously existing software needs to be evolved to ensure GDPR-compliance.

Furthermore, from a knowledge engineering and automated reasoning perspective, several ontologies for privacy requirements engineering were proposed to aid machine-intelligence use in the field; most prominently, Gharib et al. [28] provide an overview of the state of the art. In this respect, automated reasoning is still limited to capturing the requirements as opposed to aiding in the design and refactoring activities immediately stemming from such requirements. The intention of the analysis and results reported in this paper is explicitly that of filling this gap, to aid future efforts of systems (re-)design in compliance with GDPR.

Finally, focusing on GDPR, some early works have tried to distill the regulation in some design-friendly form, e.g., Renaud et al. [29] offer an outline of state of the art transition patterns onto software privacy policies which are useable yet compliant with the ruling. Similarly, Tesfay et al. [30] offer a case-study of the considerable expense required to adapt previously-existing policies onto GDPR. To help reduce that expense, the results in this paper aim to characterise further the contents, structure, features, and characteristics of GDPR and the restrictions therein.

3. Research design

3.1. Research problem and questions

This manuscript addresses the research problem of formalising the GDPR in (1) a machine-readable format allowing for mining the software design principles intrinsic in the regulation, and (2) following a rigorous method which minutely conforms to all the concepts, attributes, and implications defined in the GDPR, thus allowing for use of results in further automated software engineering research and practice.

To address both objectives (1) and (2) the manuscript synthesises Formal Concept Analysis (FCA) in the next section. In the following, the research questions behind (1) and (2) are elaborated.

- RQ1 What software design principles enable GDPR-compliance *by-design*?
- RQ2 Are there any distinguishable levels of design complexity and thus GDPR-compliance?
- RQ3 What are the most important requirements the GDPR puts forth?
- RQ4 How do the findings relate to real-world GDPR compliance cases?

Research question 1 is aimed at distilling any widely applicable laws, guidelines, biases and design considerations reflecting the accumulated knowledge and experience garnered through FCA analysis of the regulation; examples of similar design principles are the SOLID object-oriented design principles.[12] Research question 2 is aimed at understanding whether the regulation encapsulates any distinguishable and superimposed levels of design complexity, namely, any sets of concepts and requirements which depend on other requirements which must be satisfied first; in that case, the regulation would also encapsulate an implicit *sequence* of design steps which must be undertaken for compliance. Furthermore, research question 3 is aimed at understanding whether there exist concepts and requirements in the regulation which are depended upon by most others; these key requirements must be upheld first. Finally, research question 4 is intended as a *reality-check* to test the findings elicited as part of RQs 1–3 against real-life GDPR-compliance examples. The rest of this section focuses on illustrating the research method adopted to address RQs 1–3, while the recap of methods and approaches to address RQ4 is self-contained later in the evaluation section, Section 5.

[12] https://web.archive.org/web/20150202200348/http://www.objectmentor.com/resources/articles/srp.pdf.

3.2. Research method: FCA explained

This subsection details the FCA method tailoring and paraphrasing its formal description from previous work on the matter [9,10] — for the sake of simplicity, the demonstrations of theorems quoted in this section shall not be represented nor provided. The interested reader is sent to related literature — furthermore, the digest immediately below is intended for practitioners who are more interested in interpreting and using the results in action rather than fathoming their mathematical meaning.

> **Method Digest.** Formal concept analysis of GDPR makes explicit the *concept lattice* of GDPR, that is, the sum of all potentially interesting clusters of concepts and relationships hidden in the regulation.

FCA analyses data which captures relationships between a particular set of concepts and a particular set of attributes. The goal of FCA is to produce two kinds of output from the aforementioned input data. The first is a *concept lattice* — a collection of formal concepts in the data which are hierarchically ordered by means of subconcept-superconcept relations. Formal concepts are particular clusters which represent natural human-like concepts such as "organism floating in liquid", "car with roll-bar system", etc.

The second output of FCA is a collection of *attribute implications*. An attribute implication describes a particular dependency which is valid according to the data, e.g., "every number divisible by 3 and 4 is divisible by 6", "every respondent with age over 60 is retired", etc.

A distinguishing feature of FCA is an inherent integration of 3 analysis and synthesis components of conceptual processing of data and knowledge, namely, (1) the discovery and reasoning with concepts in data, (2) the discovery and reasoning with dependencies in data, and (3) the visualisation of data, concepts, and dependencies with folding/unfolding capabilities.

It should be noted that features (1)–(3) are intrinsic to FCA and match exactly all research conceptives captured in Section 3.1.

3.2.1. Basic notions

FCA starts from producing a table, known as a *cross-table*, where concepts and their logical attributes can be represented by a triplet <G,M,I> and where I is a binary relation between G and M. Elements of G are called *concepts* and correspond to table rows, elements of M are called *attributes* and correspond to table columns, and $\forall$ g $\in G$ and m $\in$ M, <X,Y> $\in$ I indicates that concept g has attribute m while <G,M> $\notin$ I indicates that "g" does not have "m".

For instance, Fig. 1 depicts a table with logical attributes. The corresponding triplet <G,M,I> is given by G = {x1, x2, x3}, M = {y1, y2, y3}, and we have $\langle x_1, y_1 \rangle \in$ I, $\langle x_2, y_3 \rangle \notin$ I, etc.

	y_1	y_2	y_3	$\cdots$
x_1	×	×	×	
x_2	×	×		⋮
x_3		×	×	
⋮		$\cdots$		$\ddots$

Fig. 1. Table of FCA concepts (rows) with logical attributes (columns) — the table is commonly known as cross-table.

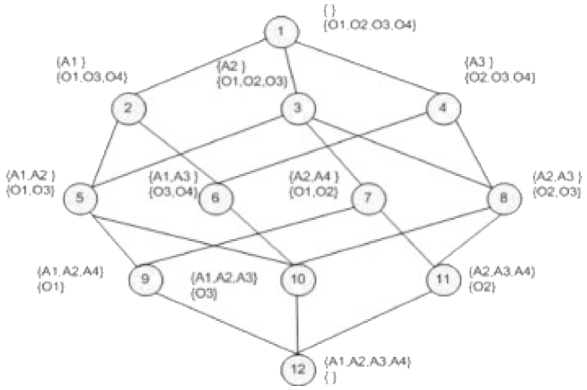

Fig. 2. Concept Lattice example, tailored from instantiating the table in Fig. 1 with more concrete concepts (Ox) and attributes (Ax).

Assuming a table such as the one represented in Fig. 1 is available, FCA aims at obtaining two further outputs. The first one, called a *concept lattice*, is a partially ordered collection of particular clusters of concepts and attributes. The second one consists of formulae, called *attribute implications* — implications describe particular attribute dependencies which are true in the reference table. The clusters, called formal concepts, are pairs $\langle A, B\rangle$ where A ⊆ X is a set of concepts and B ⊆ Y is a set of attributes such that A is a set of all concepts which have all attributes from B, and B is the set of all attributes which are common to all concepts from A. For instance, from Fig. 1, $\langle x1, x2, y1, y2\rangle$ and $\langle x1, x2, x3, y2\rangle$ are examples of formal concepts of the (visible part of) the table. An attribute implication is an expression A ⇒ B with A and B being sets of attributes. A ⇒ B is true in table <X,Y,I> if each concept having all attributes from A has all attributes from B as well. For instance, $y3 \Rightarrow y2$ is true in the (visible part of) the table in Fig. 1, while $y1, y2 \Rightarrow y3$ is not (x2 serves as a counterexample).

3.2.2. Concept lattices

Formally, as previously stated, a (cross-)table such as the one represented in Fig. 1 is represented by a so-called *formal context*.

Under the aforementioned premises, the *extent* of the concept (X, Y) is X and its *intent* is Y. The formal concepts of a context are ordered by the sub- and super-concept relations. The set of all formal concepts ordered by sub- and super-concept relations forms a *concept lattice*. Fig. 2 (tailored from related work [31]) shows the concept lattice augmented from the context in Fig. 1, where a node represents a concept labelled with its intensional and extensional description. The links represent the sub- and super-concept relations.

3.2.3. Concept-forming operators

Every formal context induces a pair of operators, concept-forming operators:

I	y_1	y_2	y_3	y_4
x_1	×	×	×	×
x_2	×		×	×
x_3		×	×	×
x_4		×	×	×
x_5	×			

Fig. 3. Table of FCA concepts (rows) with logical attributes (columns) — a Formal Concept is evidenced; the notion of a formal concept is conveniently be described as maximal rectangles in the cross-table.

Definition 2 (*Concept-forming Operators*)**.** For a formal context <X,Y,I>, operators $\uparrow : 2^X \longrightarrow 2^Y$ and $\downarrow : 2^Y \longrightarrow 2^X$ are defined for every A ⊆ X and B ⊆ Y by

A↑ = { y ∈ Y ‖ for each x *in* A : $\langle X, Y\rangle$ ∈ I }, and
B ↓ = { x ∈ X ‖ for each y ∈ B: $\langle X, Y\rangle$ ∈ I }.

Essentially, Operator ↑ assigns subsets of Y to subsets of X. A↑ is just the set of all attributes shared by all concepts from A. Dually, operator ↓ assigns subsets of X to subsets of Y, so that B ↑ is the set of all concepts sharing all attributes from B.

3.2.4. Formal concepts and concept lattices

As a result of the above definitions and operators, a formal concept F is a cluster in a cross-table, defined by means of attribute sharing. In layman's terms, formal concepts (or *objects*) can conveniently be defined as maximal rectangles in the cross-table. More formally:

Definition 3 (*Formal Concept*)**.** A formal concept in <X,Y,I> is a pair <A,B> of A ⊆ X and B ⊆ Y such that A↑ = B and B ↓ = A.

For a formal concept <A,B> in <X,Y,I>, A and B are defined as the extent and intent of <A,B>, respectively. In layman's terms formal concepts can be thought as couplings <A,B> if and only if A contains only concepts sharing all attributes from B and B contains only attributes shared by all concepts from A. For example, on the table in Fig. 3, the greyed-out area constitutes a formal concept.

Mathematically, <A,B> is a formal concept if and only if <A,B> is a fixpoint [32] of the pair $\langle\downarrow, \uparrow\rangle$ of concept-forming operators — the fixpoints in question can be generated and continuously checked algorithmically as part of FCA tool support.

Finally, the collection of all formal concepts of a given formal context, as represented by a certain cross-table, is called a *concept lattice*.

Definition 4 (*Concept Lattice*)**.** Denote by λ(X, Y, I) the collection of all formal concepts of <X,Y,I>, i.e. λ(X,Y,I) = {<A,B> ∈ 2^X x 2^Y ‖ A↑ = B , B↓ = A}. Assuming λ (X, Y, I) reflects the subconcept-superconcept ordering ≥ in <X, Y, I>, then λ(X, Y, I) is called a concept lattice of <X, Y, I>.

Essentially, λ(X, Y, I) represents the sum of all (potentially-interesting) clusters which are *hidden* in the data <X, Y, I>. Key interesting mathematical properties of such a representation are that (1) a complete lattice is a partially ordered set; (2) all subsets have a supremum (called *join-point*, or simply, join) — a supremum of a set C can be seen as the most specific formal concept that generalises all the concepts from C; (3) all subsets have an infimum (called *meet-point*, or simply, meet).

3.2.5. Analyses over concept lattices

The mathematical theory behind formal concepts allows for several analyses to be applied over cross-tables as part of FCA. First, formal contexts can be *clarified* — A formal context <X, Y, I> is *clarified* if the corresponding table does neither contain identical rows nor identical columns; it is proven that the two contexts (that is, the clarified and non-clarified one) are logically equivalent.[13]

Furthermore, in the scope of a formal context <X, Y, I>, an attribute y ∈Y is called *reducible* if and only if there is an Y' ⊂ Y with y ∉ Y' such that:

$$\{y\}\downarrow = \cap_{z \in Y'} \{z\} \uparrow$$

meaning that the column corresponding to y is the intersection of columns corresponding to z from Y'. The same axiom applies identically to concepts. An analogy with linear numbers is that If a real-valued attribute y is a linear combination of other attributes, it can be removed; in the scope of GDPR, this reduction operation does not apply, since lattice reduction depends on what is to be done with the attributes and/or concepts, which, in the case of GDPR are strictly required in their totality.

It should be noted that all the results outlined in Section 4 were inferred automatically through tools that support automated inference for FCA. None of the results showcased in this manuscript were elicited or interpreted in any way, but rather reported as such. Therefore, the results themselves are to be seen as a starting point for formal verification and compliance approaches to be applied in the scope of software systems verification and validation.

The rest of operations conducted during formal concept analysis, such as NextClosure [33,34] or non-redundancy tests [35] were left out of the analysis reported in this manuscript since they were deemed out of scope. Similarly, the basic mathematical structures computable using FCA algorithms, namely, Galois Connections [36] and closure operators [37] are left for the reader to investigate in related literature since their treatise here would be lengthy, fruitless, and discouraging.

Conversely, in the scope of this paper, the cross-table representing GDPR was elicited and used to compute the concept lattice corresponding to GDPR. Subsequently, the following basic FCA analyses were applied: (1) the denser concept lattices were inferred; (3) a Voronoi-chart was inferred from available data to understand the partitioning sub-planes of the GDPR concept plane into regions of complexity based on distance to points in the specific subset of concepts identified for the GDPR plane [38]. All analyses were supported by the ConExp tool, as detailed later in Section 3.3.2. These results, namely, the concept lattice, core implications and supreme formal concepts, are showcased in the results section (see Section 4).

3.3. Applying FCA to the GDPR

This subsection summarises the way in which the concepts and analyses outlined in Section 3.2 were applied in the context of the GDPR, also by means of selected tool-support. The basic assumption of the FCA application is that concepts and attributes appearing in the articles of the regulation are already linked by a semantic relation reflecting the regulation meaning and restrictions — assuming that this relation can be reduced to a binary relation (i.e., regulations are either upheld or violated) then an application of FCA focused on eliciting concepts and attributes to be represented over an FCA tableau is theoretically sound. The rest of this section details how juxtaposed concepts and attributes were elicited manually as part of FCA.

3.3.1. Method of application

This section provides rules and details concerning how FCA was applied to the GDPR text. Fig. 4 provides a sample comma (i.e., Article 7, comma 2 from the regulation) as encoded by means of FCA. The application of the method entails the use of two basic rules:

1. **Noun.** Concepts are generally expressed by nouns since nouns represent the basic atomic entitlements for concepts;
2. **Adjective.** Attributes are generally expressed by adjectives since adjectives enrich basic atomic concept clauses with detail, a single detail per adjective;

To the two basic rules above, previous examples of text-based FCA analysis [39,40] envision the application of an additional set of 4 rules:

3. **Phrasal Verb.** A phrasal verb is any verbal clause matching a verb with its accompanying phrasal particle (e.g., "come into" or "move within"). A phrasal verb identifies a further specification or contextualisation with respect to a concept; therefore, this rule specifies that an attribute needs to be created for every phrasal verb to be found across the reference text. In an initial version of this rule, any connecting adverbs would be added as interconnection particles for the phrasal verb in question. After Inter-Rater Reliability (IRR) assessment the aforementioned latter part was deemed too restrictive, and rephrased to account for both noun and verbal forms of verbs.
4. **Conjunction.** A conjunction is textual connection between concepts with attributes, or both. Because of their syntactical nature, no concepts or attributes should be added to an FCA cross-table. Rather, the conjunction signifies that more attributes or concepts are about to be defined further into the text.
5. **Adverb.** An adverb is a modifying clause that adds attributes to both attributes or contexts; consequently, an attribute should be created or altered whenever an adverb is found.
6. **Interconnecting Particle.** An interconnecting particle is any textual form which does not fall in the previous categories. Examples are "other", and "which". They represent neither concepts nor attributes and are therefore left out of the cross-table.

To assess the coverage of the regulation with the above-defined rules – that is, to understand whether the above text-mining rules are sufficient to effectively cover the entire regulation or which parts of it – the Naive-Bayes (NB) machine-learning algorithm (common for text mining [41]) was applied to classify the entire regulation using said rules as classes. The NB was trained with a selected subset of 70% of the regulation (prepared by applying the above rules via the online text classification application called uClassify[14]) and tested over the remaining 30%, for an accuracy of 92,3% (F-Measure). As aforementioned, the goal was to understand the ratio of the regulation left out by ignoring interconnecting particles (i.e., according to rules 1–6 defined above). This evaluation analysis revealed that 7.8% of the entire regulation is left out using the rules above — with manual inspection of this part, it was found that it relates to interconnecting text particles only.

[13] The proof is not reported here for the sake of space.

[14] https://www.uclassify.com/.

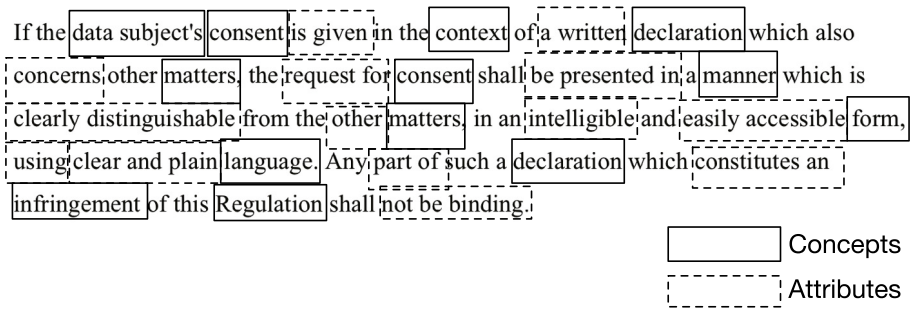

Fig. 4. Applying FCA to GDPR - a sample extract from the regulation and the application of the analysis to the text.

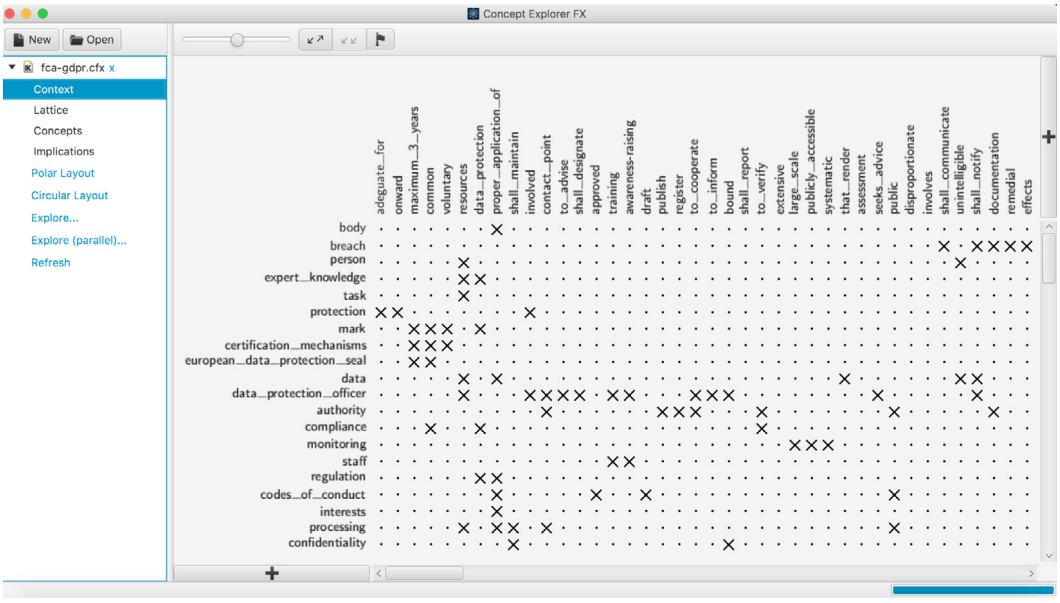

Fig. 5. GDPR FCA, cross-table extract.

In the scope of the previous experimentation Python version 2.7.3 and Jupyter notebook.[15] were used — both were used by means of the anaconda environment.[16]

Furthermore, the above textual classification experiment not withstanding, to strengthen construct, internal, and external validity, the ML experimentation thus outlined was only used to quantify the coverage achieved over by FCA — conversely, the entire FCA analysis and synthesis according to the previously-defined rules was conducted manually.

3.3.2. Tool-support

To support the semi-automated analysis of the GDPR cross-table, we used two tools available to support the experimentation by means of FCA. First, the "ConExp-NG"[17] tool was used in a GUI-centric way to explore the cross-table once obtained; the tool also allows the creation of formal contexts out of the cross-table as well as draw and visualise the concept lattices embedded within the table. Second, the "ConExp-FX"[18] tool was used to provide for advanced visualisations and implicational processing of the cross-table at hand. These features of the tools were also used to explore hidden dependencies between top-recurring GDPR attributes. Fig. 5 outlines an extract of the cross-table for GDPR as represented in the ConExp-FX environment.

3.4. Inter-rater reliability assessment

The preparation of the cross-table upon which the FCA method is based still heavily relies on human interpretation; though made rigorous as much as possible (see Section 3.3), such interpretation is still subject to observer bias which must be assessed and evaluated to establish the so-called Inter-Rater Reliability (IRR) assessment [42]. In the scope of this exercise, the well-known Krippendorf α coefficient was employed [43]. More in particular, to apply the Kα IRR score, a randomly-selected sample of 15% of the total amount of pages from the GDPR document were isolated to be re-analysed by an independent third-party researcher, solely to the purpose of generating a cross-table for that selected sub-portion according to the procedures specified in the previous sections. The cross-table for the same sub-portion was re-generated independently by the author of this manuscript and inter-rater reliability assessment was conducted. An initial value of K_α of 0.71 led to discuss and fine-tune the procedure summarised in Section 3.3. More in particular, Rules n. 3 and 5 were fine-tuned to drop connecting adverbs and particles as well as to account both noun and verbal forms of verbs in the GDPR text. Subsequently to the adjustments, the IRR assessment exercise was run again and amounted to $K_\alpha = 0.96$, higher than the standard reference value of 0.800.

15 http://jupyter.org/.

16 https://www.nsc.liu.se/systems/triolith/software/triolith-software-apps-python-2.7.13-anaconda-4.4.0.html.

17 https://github.com/fcatools.

18 https://lat.inf.tu-dresden.de/~francesco/_previous_version/conexp-fx.html.

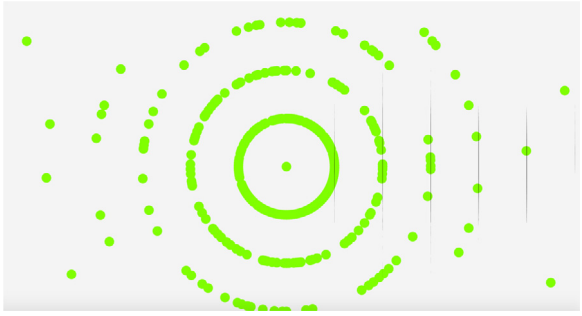

Fig. 6. GDPR FCA, radial diagram outlining identified concepts and attributes with their implicational distance from the supremum (centre of the plot); 4 implicational levels exist.

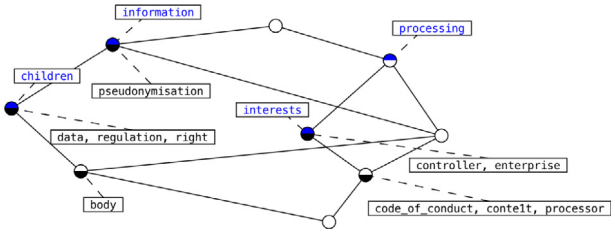

Fig. 7. GDPR FCA, a formal context connecting the regulation's focus on children and the processing controllers' obligations.

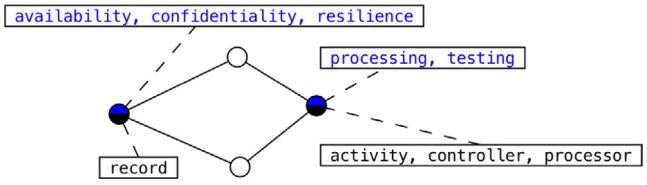

Fig. 8. GDPR FCA, a formal context connecting the obligations of controllers and processors over record processing.

4. Results

This section first outlines the visual and statistical results of FCA applied to GDPR. Subsequently, the section describes the denser formal concepts in the regulation, as well as outlining the core implications of the regulation.

4.1. Visual and statistical results

FCA revealed a total of 144372 formal concepts, elicited from in 143 concepts and 318 attributes — a snapshot of the regulation cross-table is outlined in Fig. 5. The radial diagram in Fig. 6 highlights the existence of 4 implicational levels intrinsic to GDPR — each level corresponds to its own concept lattice and constitutes a degree of complexity or *level of compliance* intrinsic to the regulation; starting from the supremum and browsing each lattice from the innermost circles, software designers can redesign or refactor their own systems working their way towards the most external circles, using the concepts and attributes therein as dimensions and descriptors of the stakeholders, concerns, risks, and constraints specified in the regulation. As part of this refactoring exercise, designers shall start from the outer-most circle, since that circle identifies concepts that imply other concepts and shall therefore be considered first. Subsequently, the other circles can be addressed incrementally, e.g., following incremental and iterative systems maintenance and evolution practices [44]. As another example, consider the scenario wherefore a previously existing system needs to be assessed with respect to the regulation; in that case, designers can start mapping the system requirements onto the 4 implicational levels identified and reported in this article to understand if and how the current system structure, properties, and configuration complies to the regulation.

4.2. RQ1: Formal contexts

To address RQ1, namely, "*What software design principles enable GDPR-compliance by-design?*", the concept lattice for the regulation was analysed to perform *strongest formal-context mining* [45,46]. More specifically, automated tool-support for FCA can be used to identify, using association-rule mining [47], the formal contexts with the widest set of attributes and formed by the widest set of concepts. As a result of this process, 3 formal contexts with higher support (i.e., Support is an indication of how frequently a discovered itemset appears in the dataset, it is a well-known evaluation metric for association-rule mining [48]).

Figs. 7 to 9 outline the formal concepts thus identified. The figures are to be read from the most connected concepts (i.e., concepts whose representation node is fully-coloured) to the most-immediately implied concepts (i.e., concepts or attributes whose representation node is coloured in the top part) to the end-implications (i.e., concepts or attributes whose representation node is coloured in the bottom part) - finally, empty nodes imply the existence of an implicit concept which was not identified in the regulation or was that is implied by semantics. The concepts in question elaborate on: (FC1) the regulation's focus on *children*; (FC2) the obligation of *controllers/processors*; (FC3) the role of *data protection officers*.

First, FC1 outlines a formal context that details the controllers obligations which relate to the protection over *processing* of *information* for *children's* data. The information in question should be protected with apposite mechanisms (e.g., pseudonymisation) and established through a *code of conduct* (e.g., a privacy policy) with reference to a specific *context* (e.g., market analysis) and a specific *processor* (e.g., an analytics and profiling dashboard and its responsible organisation). The regulation also implies that the *interests* of the controller or enterprise body enacting the processing should be made appropriately explicit for the natural people in question — that is, in the scope of this context which relates to children, the natural people equate to the responsible parties to oversee on the children in question.

As a design principle, this formal context reflects the need to design specific support for processing of children data and, in general, support to codes-of-conduct over multiple data-processing levels.

Second, FC2 outlines a formal context that details the *controllers* and *processors*' obligations over processing of *records* as well as their obligations to provide proof of evidence of records' *availability*, *confidentiality*, processing *resilience*. Finally, the obligations rest on offering all necessary facilities for *testing* and making evident all the aforementioned properties for processing systems and software.

As a design principle, this formal context reflects the need to (a) design and (b) test for more explicit user-driven control over what data can be processed and how.

Third, FC3 outlines a formal context that highlights the *purpose* of the *data protection officer*, who is entrusted with making evident, eliciting, defending, and maintaining the *basis* for the *lawfulness* of the *processing* conducted by a processing *controller* and under its own *interests*. The basis in question is associated to a specified processing *period* and over specified data *recipients*.

This formal context defines the need to design for the implicit organisational structure intended in GDPR, rotating around the

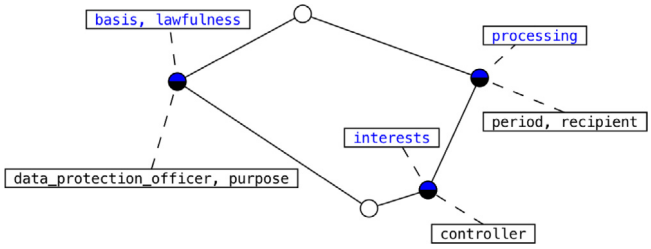

Fig. 9. GDPR FCA, a formal context elaborating further on the key concepts around the role of data protection officers in the context of the regulation.

data protection officer. The design principle that can be distilled from this is the need to design explicit support for the *processors* and the responsible parties in them, that is, the data protection officers. Explicit software design shall be dedicated to their role and the extent to which they interact with the processing software.

Summary for RQ1 — design principles for GDPR-compliance.

1. *design for controllable interaction* — appoint a data protection officer and restructure software designs in line with their technical needs and organisational interaction demands;
2. *design for age-based data-processing* — distinguish design, implementation, and testing phases explicitly between data processing for children and data processing for adults, allowing for appropriate rules and regulations for each, based on the appointed responsible party in either case;
3. *design for accountability* — devise the design to make every part of the processing accountable to both human, cyber-, and physical assets involved in the processing; prepare organisational codes-of-conduct to increase awareness, training, and consistency of GDPR-compliance under the responsibility of the data protection officer and re-design software solutions to support such codes.

4.3. RQ2: Dimensions of design complexity

To address RQ2, namely, "*Are there any distinguishable levels of design complexity and thus GDPR-compliance?*", the concept lattice of the regulation was processed by applying automated Voronoi partitioning [49,50] as outlined in Section 3. In mathematics, a Voronoi tessellation [50] is a partitioning of a plane into subregions using a distance function featuring specific given points in a specific subset of the plane. That set of points (called seeds) is specified beforehand, and for each seed there is a corresponding region (called a Voronoi Cell) consisting of all points closer to that seed than to any other. In the case of GDPR and its FCA, Voronoi partitions identify cliques of concepts (and attributes) that need explicit and individual addressing, and therefore can be used as a rough estimate of design complexity for GDPR compliance.

To conduct this analysis over the regulation, the concept lattice was partitioned using the well-established convex distance function known as the Minkowski distance as outlined in previous reference work in the context of Voronoi partitioning [51] — the result of this analysis revealed the existence of 27 Voronoi partitions across the concept space defined by the regulation [49].

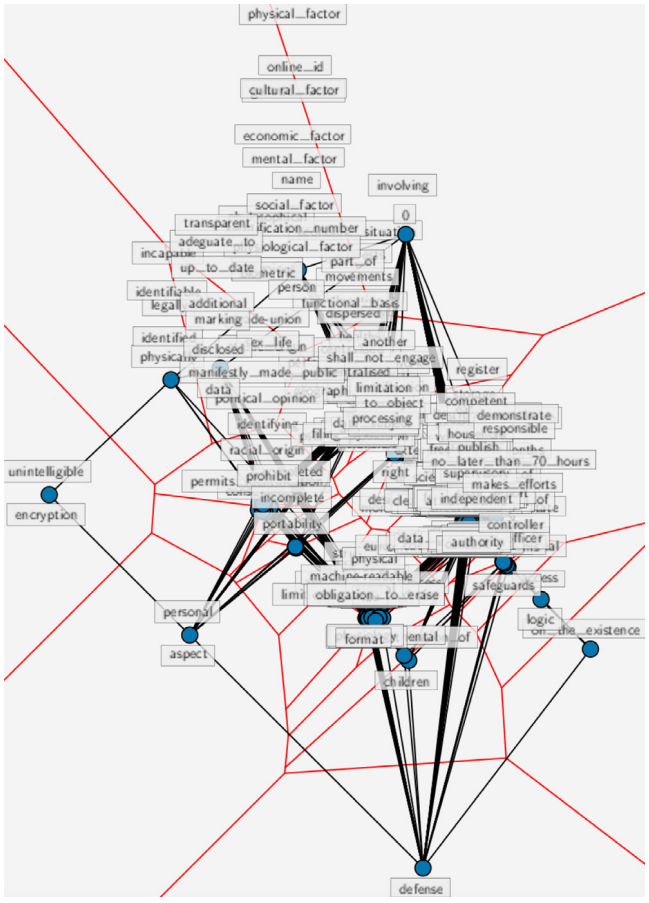

Fig. 10. GDPR FCA, Voronoi partitions existing across the conceptual space; 27 partitions exist.

Fig. 10 outlines the results of this analysis. The figure shows the 27 Voronoi partitions existing across GDPR concept lattice. This information allows understanding the depth of complexity of the regulation, e.g., for further processing of the FCA data elicited in this study by means of Machine-Learning techniques [52] beyond the partitioning mechanisms used in this manuscript.

The major implication from the perspective of software and information systems design is that, assuming that each concept in the regulation requires explicit requirements engineering and design, then the design space to uphold every requirement explicitly stated in the regulation can be arranged into no less than 27 distinct design-rule spaces (i.e., series of concepts and attributes which need explicit software design rules).

Further research should be invested to understand whether these 27 spaces constitute *design-rule spaces* [53] emerging in correspondence to those subsets.

Summary for RQ2. There exist 27 Voronoi partitions across the regulation; these partitions constitute disjoint design spaces which need explicit addressing. Further research must be used to establish the extent to which these spaces can be used to represent *design-rule spaces* [53].

4.4. RQ3: Core implications

To address RQ3, namely, *What are the most important requirements the GDPR puts forth?*, the resulting FCA matrix was analysed

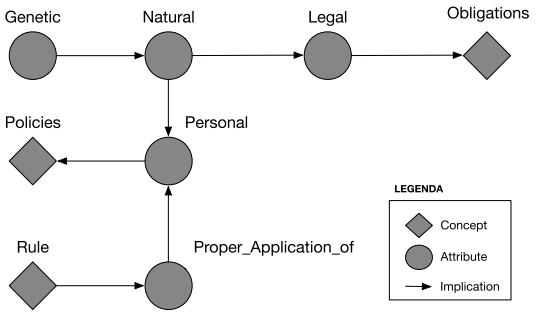

Fig. 11. GDPR FCA, core implications (arrows) among core concepts (side-squares) and attributes (circles) extracted from the regulation.

by means with association-rule mining. The association-rules extracted with tool support are outlined in Fig. 11. The figure shows that there exist 7 rules behind the regulation, namely:

1. *Genetic* ⇒ *Natural*.
2. *Natural* ⇒ *Legal*.
3. *Legal* ⇒ *Obligations*.
4. *Natural* ⇒ *Personal*.
5. *Personal* ⇒ *Policies*.
6. *Rule* ⇒ *ProperApplicationOf*.
7. *ProperApplicationOf* ⇒ *Personal*.

Combining the above rules 3 essential constraints can be distilled from the regulation; concepts or attributes from the regulation appear in *italics*.

First, the implications above make clear the relation between natural and legal entities involved in data processing. On one hand, the natural entity (i.e., a subject which can be associated with *personal* genetic information) is referred to as the natural person or *data subject* who is at risk of withstanding a violation of data protection rights. On the other hand, the legal entity has *obligations* with respect to the natural person to a number of further concepts (e.g., informed consent, right to withdrawal of permission, right to information, right to deny processing and more).

Second, to the extent to which the data subject expresses consent for processing of his/her own data, the *policies* inside GDPR are designed to act on the *personal* data, that is, the data which can be directly associated to the specific racial, genetic, biological, individual, social, or otherwise nature of the processed individual.

Third, finally, the *rules* inside the regulation ensure the *proper application* of *personal* data protection *policies*.

Summary for RQ3. There exist seven association rules with high support across the regulation which must be addressed during software and information systems design for GDPR-compliance; the implications of the rules in question reflect the distinction between natural, legal, personal, and genetic data processing as well as the proper/verifiable and accountable application of processing to such data. This finding reinforces the aforementioned *design for accountability* design principle.

5. Evaluation

As previously mentioned, the results of the FCA analysis were tested against two real-life industrial case-studies. This section reports on the method followed to design and execute said case-studies as well as the results thereto.

5.1. RQ4: Research methods

The two cases in object reflect two different observational studies enacted in a large service provider headquartered in the Netherlands known as BDO; the organisation details are reported below.

BDO is the fifth largest accounting network worldwide, with over 1,000 offices in more than 150 countries. They provide services in the fields of accounting, tax, advisory and consulting. BDO clients range from small, medium-sized enterprises (SMEs) to multinationals. Their core activity is to provide professional services to their clients. BDO client-centred approach helps companies to achieve their goals offering tailor-made solutions to achieve the highest quality when it comes to their services. BDO has four Lines of Services (LoS), the IT audit department of BDO which is part of the Audit and Assurance LoS that accompanies this study.

In the scope of this study, BDO wants to devise a framework for GDPR-compliance to be used as part of its service offer, with a focus on (case 1) GDPR-compliance towards healthcare information systems and (case 2) GDPR-compliance in public governance information systems. More specifically, the design framework in question is aimed at supporting the design, implementation, and operation of information systems to be deployed and operated in the scope of cases 1 and 2.

The objective of the evaluation showcased here is to show that the aforementioned framework, as constituted by roles, design challenges, and design constructs, matches the design principles and challenges showcased previously in Section 4.

5.1.1. Data acquisition

in the scope of both case-studies, a total of 37 expert practitioners with an average of 5+ years experience (std. dev. 1,2) in designing, implementing, and operating privacy-aware information systems were interviewed with a semi-structured approach. The approach structured the interview process by rotating around 3 topics to be discussed in sequence, namely: (1) roles whose software support is essential for GDPR compliance stemming from the case-studies; (2) design challenges which are critical to be addressed for such compliance in the scope of the studied cases; (3) any specific design constructs requiring special attention from software and information systems architects and observable in the cases in object. In the scope of the three phases, subjects were encouraged to make anonymised examples of GDPR principles being taken as design concerns as well as design principles being used as requirements. Finally, the practitioners were presented with the design principles recapped in Section 4 and asked to map said principles with their own design challenges, providing examples where appropriate.[19]

Data acquisition started on Feb. 2018 and proceeded over a 3-month interview period — the period in question reflects the bootstrap of late-stage prototyping and principal implementation of the systems in the scope of cases 1 and 2.[20]

5.1.2. Data analysis and synthesis

All interviews were transcribed and coded following a simplistic content analysis and thematic coding [43]. Available data stemming from the content analysis was loaded into tables formatted to be analysed with the R data analytics toolkit by means of the two students involved in the case-studies in object. At

[19] An overview of the mapping for this final exercise is provided online: https://tinyurl.com/GDPR-FCA-Mapping.

[20] Descriptive details for both cases are protected by Non-Disclosure Agreements and cannot be disseminated beyond the details currently contained in this manuscript.

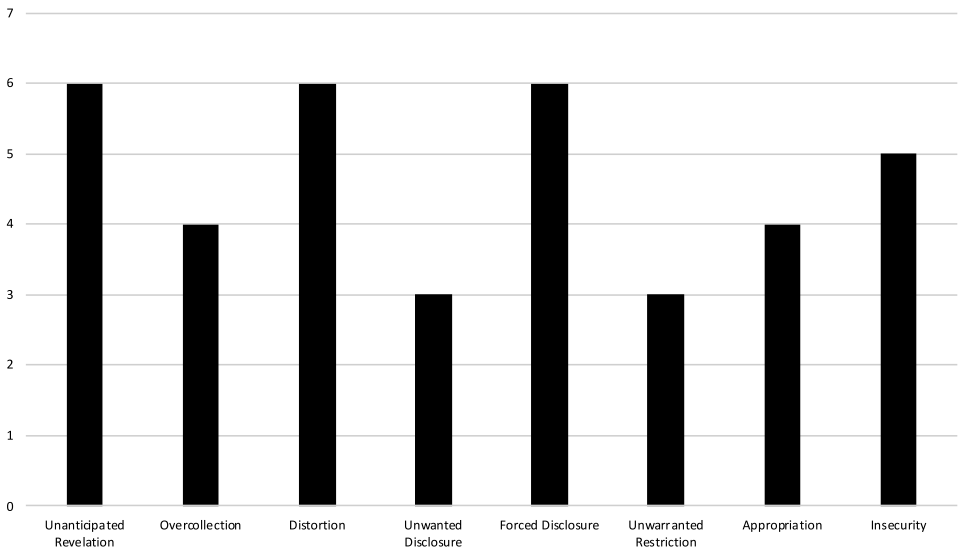

Fig. 12. Case-study content analysis, challenges relative frequency normalised per participants number.

this point, observer reliability was assessed, thus evaluating the agreement across analyses over the data tables produced. Subsequently, we computed the Krippendorff's α coefficient for observation agreement [43] - the α score essentially measures a confidence interval score stemming from the agreement of values across two distinctly-reported observations about the same event or phenomenon. In our case the value was applied to measure the agreement between configuration details, analyses, and statistical results of our analysis. The value was calculated initially to be 0.83, hence $\alpha > .800$, which is a standard reference value for highly-confident observations. Subsequently, the value was used to drive the agreement between the two analyses up towards total alignment.

Subsequently, card-sorting and taxonomy analysis approaches [54] were used to conduct content and frequency analysis of the challenges emerging from both case-studies.

5.2. Evaluation results

Fig. 12 showcases the frequency of challenges reported by the participants while Table 1 showcases resulting themes (Col. 1) and design challenges (Col. 2) as part of the case-studies evaluation, with an elaboration of the GDPR clause violation reported by the interviewees (Col. 3) and the recommended mapping with design principles from this article aggregated by frequency (Col. 4).

The table confirms a considerable importance of design principle 1 proposed in Section 4, namely, the *Design for controllable interaction* principle. Noticeably, the principles always appear conjoined – with the exception of challenges connected to data disclosure under specific scenarios – indicating a compounding interaction of the challenges connected to all 3 principles all across the design space reflected by the observed cases. Finally, the dimensions of Table 1 were mapped to a conceptual framework for quick reference to practitioners; the framework is reported in Fig. 13.

In summary, the following evaluation of the insights gathered through FCA can be given.

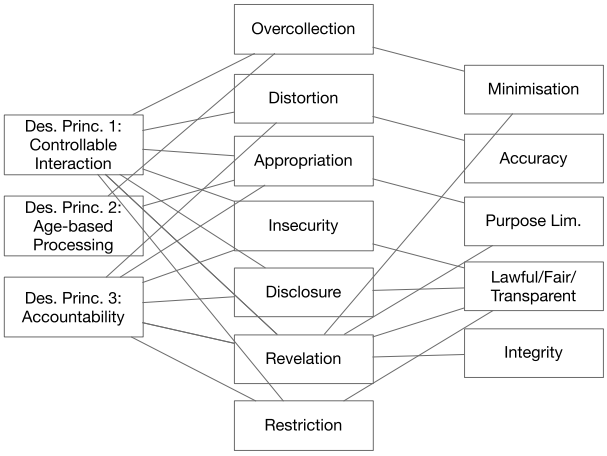

Fig. 13. Mapping real-life evidence (right-hand side) of GDPR-related design challenges with FCA-based design insights (left-hand side); a conceptual overview.

Summary for RQ4. There is consistent evidence towards the importance of design principles 1, 2, and 3 in the scope of implementing and deploying GDPR-consistent information systems; evidence suggests that the controllable interaction principle, namely design principle 1, is more impactful than the remaining two in the scope of the systems evaluated within the cases observed by this study.

6. Discussion

This section discusses results with a number of observations made while analysing GDPR-FCA data and results from the previous section, as well as lessons learned in the process of preparing, analysing, synthesising, and discussing such data.

6.1. Observations: Re-designing for GDPR

The findings outlined in the previous section lead to at least 3 observations.

Table 1
Real-life industrial design challenges (col. 1) for GDPR-compliance, with a description (col. 2), violated articles risk (col. 3) and mapping to design principles from this paper (col. 4).

Design Issue	Description	Violated GDPR clauses	P#
Overcollection	Collecting excessive amounts of data or otherwise unjustifiable collection of data over different types of subjects.	Data minimisation principle	1,2
Distortion	Using or disseminating inaccurate or out of date personal data reflecting all policy levels.	Accuracy principle	1,3
Appropriation	Using data beyond what data subjects could reasonably expect.	Purpose limitation principle	1,2,3
Appropriation	Using data in a manner any reasonable individual would object to in this context.	Lawfulness, fairness and transparency principle	1,3
Appropriation	Using data to make unjustifiably decisions, which the organisation cannot objectively defend.	Lawfulness, fairness and transparency principle	1,3
Insecurity	Data getting lost/stolen or data being destroyed/altered.	Integrity and confidentiality principle	1,3
Unwanted disclosure	Data being unjustifiable or accessed unauthorised, transferred, shared or published.	Integrity and confidentiality principle	1
Forced disclosure	Data subjects feel they are pressured to share personal data. Forced disclosure can occur when data subjects feel forced to provide information disproportionate to the purpose or outcome of the process. Induced disclosure can include the threat of denying access to an essential (or perceived essential) service.	Data minimisation principle; lawfulness, fairness and transparency principle	2
Unanticipated revelation	Unexpected revelations about a data subject that can occur when combining and analysing large and/or diverse data sets (e.g., from diverse age-groups, profiles, etc.).	Purpose limitation principle	1,2
Unwarranted restriction	Unjustifiable restriction of access to personal data which includes the physical blocking of access and limiting the knowledge of the existence or use of the information.	Lawfulness, fairness and transparency principle	1,3

First, systems and software should be re-designed to make explicit all necessary information for the role of the data protection officer to carry out its duties, this includes, but is not limited to: (a) making constantly explicit to data officers the processing intent and purposes; (b) ensuring that data officers are capable of releasing any data and any inferred result connected to the processing of that data upon immediate request by any given data subject involved; (c) simplify the communication of any consequential information inferred from processed data to the involved data subjects. It should be noted that the term ''re-design'' is used loosely here to indicate any process that modifies a systems' architecture and infrastructure for compliance to the GDPR.

Second, systems and software should be re-designed without prejudice especially with respect to the processing of information relating, in any way, shape, or form, to data of children, their profile, biological, racial or otherwise specified nature.

Third, systems and software should be re-designed to make the use of privacy-enhancing technologies as well as security-enhancing approaches or middleware parametric, enacted only wherefore the evidence of their application makes explicit its effectiveness to interested data subjects.

6.2. Lessons learned

First, FCA is a powerful analysis tool but there is very little scalable and efficient tool-support for it. It is remarkable to see that in the era of Big Data several tools do exist but fail to address the gap between current tooling and the necessary processing strength to address the current needs for FCA, e.g., for social and societal information processing or for data governance. In the future, further research around providing more effective and scalable solutions for formal concept analysis shall be encouraged, in the hope of providing a working prototype for processing formal concept analysis cross-tables at large-scale and in a semi-automated, algorithmically-assisted fashion.

Second, the world of FCA is much deeper than the analyses conducted in the scope of this paper. What is more, research in the mathematically-proven theory behind lattices is ongoing and thriving — in that respect, further research stemming from the results and open data prepared and released for the replication of this study shall be encouraged, e.g., to apply later algorithms, visualisations, and analyses over GDPR so that further benefits for GDPR-savvy practitioners may ensue.

Third, although the analyses and results outlined in the scope of this manuscript offer a basis to prioritise software designs more in line with GDPR, it should be noted that there is much more within the GDPR that was not covered herewith. The materials and results are in fact to be intended as a prioritised starting point for practitioners in companies and organisations to show compliance adaptation campaigns being enacted in their respective software designs. The starting point offered in the previous pages can be further explored by progressing in the use of FCA analyses by means of this manuscript's replication package and FCA tableau provided therein.

6.3. Threats to validity

As per any empirical software engineering endeavour, the analysis and synthesis recapped in this manuscript suffers from threats to validity, identified in the scope of this work as rotating mainly around **Construct Validity** [55]. More in particular, the interpretation of constructs within the GDPR required application of observational rules to the regulation in order to elicit concepts and attributes therein. Although the systematic analysis of the regulation was conducted with firmly-established synthesis rules (see Section 3) from the state-of-the-art, their application is subject to observation bias. As previously mentioned in the same section, an extensive inter-rater reliability campaign was enacted to ensure that the constructs elicited and analysed as part of this study were systematically synthesised in a reliable fashion. Nevertheless, observation bias remains and shall be further investigated in the scope of future work. More in particular, an systematic investigation by means of machine-learning techniques partially outlined in the methods section (Section 3), will be employed to replicate the entire study results in order to achieve a new tableau of the regulation cross-table, with the goal of confirming its isomorphism with respect to the tableau employed to attain results in Section 4.

6.4. Replication package

To encourage the confirmatory study above, as well as further research, practice and other endeavours around the GDPR as synthesised in the scope of this manuscript, an extensive replication package was prepared and made available online.[21]

7. Conclusions and future work

This manuscript details the results of applying formal concept analysis (FCA) to the General Data Protection Regulation (GDPR), a formal law enforcement that describes the conformance clauses and rules to adhere to when processing data across the EU and for subject residents of the union.

The goal of applying FCA was to elicit key insights to drive or support the work of software engineers and designers into their campaign for systems and software re-design in line with the regulation. With the term "re-design" this article intends any process aimed at improving a software architecture, its properties or the infrastructure upon which that architecture operates to strengthen its compliance with the regulation.

Towards the above general goal, the application of FCA revealed many key insights about the regulation, among which: (1) the regulation supremum — meaning, its 'entry-point', to which all other elements are directly or indirectly connected — is "defence", which clearly highlights the objective of the regulation; (2) the regulation contains an organisational structure for compliance rotating around data-protection officers; (3) the regulation shows 4 degrees of complexity, which may equate to 4 levels of compliance — practitioners can use this insight to prioritise their redesign and refactoring for compliance campaigns.

In the future a replication of this study by more automated means is planned. The goal of such as study is confirming that the manually-generated cross-table for GDPR, which was also confirmed through appropriate IRR, is identical or *isomorphic* to the machine-generated alternatives.

In addition, a reiteration of the results attained and recapped in this paper is planned, by conducting a more systematic investigation of the further analyses entailed by formal concept analysis and their feasibility in the context of further analysing the GDPR.

Finally, further exploration of the regulation by means of the general software engineering community is encouraged, either through other FCA-based analyses or otherwise, by providing an extensive replication package online.[22]

Acknowledgements

The author's work is partially supported by the European Commission grants no. 787061 (H2020), ANITA, no. 825040 (H2020), RADON, no. 825480 (H2020), SODALITE. Furthermore, the author would like to thank Mrs. Pauline van de Voorde and Mr. Wilko van den Meuelen for their support in the case-studies reported in this work. Finally, the author acknowledges the fundamental contribution of Dr. Aldo Masciantonio in furthering the availability of the materials in this manuscript online.

Appendix. Web site description

Page name	Description	link
Home page	Aim of the study	gdpr-fca.github.io/index.html
GDPR	Description of the paper	gdpr-fca.github.io/paper.html
GDPR	Dataset Download	https://drive.google.com/open?id=1dyssSxTvzoheSA3LSAelG7IoM_qjG75F

References

[1] J. Hemerly, Public policy considerations for data-driven innovation, Computer 46 (6) (2013) 25–31.
[2] J. Zerlang, GDPR: a milestone in convergence for cyber-security and compliance, Netw. Secur. 2017 (6) (2017) 8–11.
[3] C. Tankard, What the GDPR means for businesses, Netw. Secur. 2016 (6) (2016) 5–8.
[4] S. Wachter, B.D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, 2017, CoRR arXiv:abs/1711.00399.
[5] S. Gürses, C. Troncoso, C. Diaz, Engineering privacy by design, Comput. Priv. Data Prot. (2011).
[6] M. Langheinrich, Privacy by design, in: G. Abowd, B. Brumitt, A. Shafer (Eds.), UBICOMP 2001, Springer, 2001, pp. 273–291.
[7] M.A. Jackson, Principles of Program Design, Academic Press, 1975.
[8] L. Bernstein, Software engineering design principles for ultra-large-scale systems, ACM SIGSOFT Softw. Eng. Notes 37 (4) (2012) 8–9.
[9] R. Wille, Formal concept analysis as mathematical theory of concepts and concept hierarchies, in: Formal Concept Analysis, 2005, pp. 1–33.
[10] B. Ganter, G. Stumme, R. Wille, Formal concept analysis, foundations and applications, in: Formal Concept Analysis, vol. 3626, Springer, 2005.
[11] D. Schlimm, On the creative role of axiomatics. The discovery of lattices by Schröder, Dedekind, Birkhoff, and others, Synthese 183 (1) (2011) 47–68.
[12] P. Valtchev, R. Jãschke, Formal Concept Analysis, in: Lecture Notes in Artificial Intelligence, vol. 6628, Springer, Berlin/Heidelberg, 2011.
[13] R.K. Yin, Case Study Research: Design and Methods, third ed., Sage Publications, 2003, URL: http://www.worldcat.org/isbn/0761925538.
[14] L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, Addison Wesley, 1998.
[15] P.C. Jorgensen, Software Testing: A Craftsman's Approach, third ed., AUERBACH, 2008.
[16] L.I. Mancini, Formal Verification of Privacy in Pervasive Systems (Ph.D. thesis), University of Birmingham, UK, 2015, British Library, EThOS.
[17] M. Rohr, A. van Hoorn, J. Matevska, N. Sommer, L. Stoever, S. Giesecke, W. Hasselbring, Kieker: Continuous monitoring and on demand visualization of Java software behavior, in: Proceedings of the IASTED International Conference on Software Engineering 2008, ACTA Press, 2008, pp. 80–85.
[18] H.v. Rossum, Privacy-Enhancing Technologies: The Path to Anonymity, Registratiekamer, Den Haag, 1995.

21 https://tinyurl.com/y8352nrv.

22 gdpr-fca.github.io.

[19] V. Diamantopoulou, K. Angelopoulos, M. Pavlidis, H. Mouratidis, A metamodel for GDPR-based privacy level agreements, in: C. Cabanillas, S. Espana, S. Farshidi (Eds.), ER Forum/Demos, in: CEUR Workshop Proceedings, vol. 1979, CEUR-WS.org, 2017, pp. 285–291.
[20] H. Nissenbaum, Privacy as contextual integrity, Washington Law Rev. 79 (1) (2004) 119–158.
[21] A. Barth, A. Datta, J.C. Mitchell, H. Nissenbaum, Privacy and contextual integrity: framework and applications, in: 2006 IEEE Symposium on Security and Privacy, 2006, pp. 15–198, http://dx.doi.org/10.1109/SP.2006.32.
[22] Q. Ni, E. Bertino, J. Lobo, S.B. Calo, Privacy-aware role-based access control, IEEE Secur. Priv. 7 (4) (2009) 35–43.
[23] B. Carminati, E. Ferrari, J. Cao, K.L. Tan, A framework to enforce access control over data streams, ACM Trans. Inf. Syst. Secur. 13 (3) (2010) 28:1–28:31.
[24] H. Ulusoy, M. Kantarcioglu, E. Pattuk, K. Hamlen, Vigiles: Fine-grained access control for mapreduce systems, in: 2014 IEEE International Congress on Big Data.
[25] A.K. Massey, P.N. Otto, L.J. Hayward, A.I. Anton, Evaluating existing security and privacy requirements for legal compliance, Requir. Eng. 15 (1) (2010) 119–137.
[26] J.C. Maxwell, A.I. Anton, Developing production rule models to aid in acquiring requirements from legal texts, in: RE, IEEE Computer Society, 2009, pp. 101–110.
[27] E. Yu, L. Cysneiros, Designing for privacy and other competing requirements, in: 2nd Symposium on Requirements Engineering for Information Security, SREIS' 02, Raleigh, North Carolina, 2002, URL: http://www.cs.toronto.edu/pub/eric/SREIS02-Priv.pdf.
[28] M. Gharib, P. Giorgini, J. Mylopoulos, Ontologies for privacy requirements engineering: A systematic literature review, 2016, CoRR arXiv:abs/1611.10097.
[29] K. Renaud, L.A. Shepherd, How to make privacy policies both GDPR-compliant and usable, 2018, CoRR arXiv:abs/1806.06670.
[30] W.B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, J. Serna, Privacyguide: Towards an implementation of the EU GDPR on internet privacy policy evaluation, in: R.M. Verma, M. Kantarcioglu (Eds.), IWSPA@CODASPY, ACM, 2018, pp. 15–21, URL: http://dblp.uni-trier.de/db/conf/codaspy/iwspa2018.html#TesfayHNKS18.
[31] G. Fu, FCA based ontology development for data integration, Inf. Process. Manage. 52 (5) (2016) 765–782.
[32] K. Ishida, Y. Shidama, Fixpoint theorem for continuous functions on chain-complete posets, Formal. Math. 18 (1–4) (2010) 47–51.
[33] R. Belohlávek, V. Vychodil, Closure-based constraints in formal concept analysis, Discrete Appl. Math. 161 (13–14) (2013) 1894–1911.
[34] J. Baixeries, J.L. Balcázar, Characterization and armstrong relations for degenerate multivalued dependencies using formal concept analysis, in: B. Ganter, R. Godin (Eds.), ICFCA, in: Lecture Notes in Computer Science, vol. 3403, Springer, 2005, pp. 162–175, URL: http://dblp.uni-trier.de/db/conf/icfca/icfca2005.html#Balcazar05.
[35] Z. Xie, Phi-generalized concept lattice models: From power concepts to formal concepts, and then to robust concepts, 2006.
[36] R. Pöschel, Galois connections for operations and relations, in: Galois Connections and Applications, vol. 565, 2004, pp. 231–258.
[37] R. Belohlávek, V. Vychodil, Formal concept analysis with constraints by closure operators., in: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Proceedings of the 14th International Conference on Conceptual Structures, ICCS 2006, in: Lecture Notes in Computer Science, vol. 4068, Springer, 2006, pp. 131–143.
[38] M. Erwig, The graph Voronoi diagram with applications, Networks 36 (3) (2000) 156–163, http://tinyurl.com/43uv6e – last visited 20th May 2008.
[39] J. Ducrou, DVDSleuth: A case study in applied formal concept analysis for navigating web catalogs, in: U. Priss, S. Polovina, R. Hill (Eds.), Proceedings of the 15th International Conference on Conceptual Structures, ICCS 2007, in: Lecture Notes in Artificial Intelligence, vol. 4604, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 496–500.
[40] B. Ganter, R. Wille, Applied lattice theory: Formal concept analysis, 1997.
[41] S. Eyheramendy, D.D. Lewis, D. Madigan, On the naive Bayes model for text categorization, 2003.
[42] K. Gwet, K. Gwet, Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity, Stat. Methods Inter-Rater Reliab. Assess. 2 (2002).
[43] K. Krippendorff, Content Analysis: An Introduction to Its Methodology, second ed., Sage Publications, 2004.
[44] G. Zhang, L. Shen, X. Peng, Z. Xing, W. Zhao, Incremental and iterative reengineering towards software product line: An industrial case study, in: 2011 27th IEEE International Conference on Software Maintenance, ICSM, 2011, pp. 418–427, http://dx.doi.org/10.1109/ICSM.2011.6080809.
[45] P.W. Eklund, Information retrieval using formal concept analysis, 2003, SITACS Seminar at the UoW.
[46] Y.H. Chen, Y.Y. Yao, Formal concept analysis based on hierarchical class analysis, in: Proceedings of the 4 Th IEEE International Conference on Cognitive Informatics, ICCI 05, 2005, pp. 285–292.
[47] B. Liu, W. Hsu, Y. Ma, Integrating classification and association rule mining, in: Knowledge Discovery and Data Mining, 1998, pp. 80–86.
[48] R. Rastogi, K. Shim, Mining optimized association rules with categorical and numeric attributes, IEEE Trans. Knowl. Data Eng. 14 (1) (2002) 29–50.
[49] R.F. Reitsma, S. Trubin, E.N. Mortensen, Weight-proportional space partitioning using adaptive voronoi diagrams, GeoInformatica 11 (3) (2007) 383–405.
[50] I. Lee, J. Yang, Voronoi-based topological information for combining partitioning and hierarchical clustering, in: CIMCA/IAWTIC, IEEE Computer Society, 2005, pp. 484–489, URL: http://dblp.uni-trier.de/db/conf/cimca/cimca2005-2.html#LeeY05.
[51] L.P. Chew, R.L.S. Dyrsdale III, Voronoi diagrams based on convex distance functions, in: Proceedings of the First Annual Symposium on Computational Geometry, SCG '85, ACM, New York, NY, USA, 1985, pp. 235–244, http://dx.doi.org/10.1145/323233.323264.
[52] D.G. Heath, S. Kasif, The complexity of finding minimal voronoi covers with applications to machine learning., Comput. Geom. 3 (1993) 289–305, URL: http://dblp.uni-trier.de/db/journals/comgeo/comgeo3.html#HeathK93.
[53] L. Xiao, Y. Cai, R. Kazman, Design rule spaces: a new form of architecture insight, in: P. Jalote, L.C. Briand, A. van der Hoek (Eds.), ICSE, ACM, 2014, pp. 967–977, URL: http://dblp.uni-trier.de/db/conf/icse/icse2014.html#XiaoCK14.
[54] V. Kodaganallur, S. Shim, Analysis patterns: A taxonomy and its implications, IS Manage. 23 (3) (2006) 52–61.
[55] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: an Introduction, Kluwer Academic Publishers, Norwell, MA, USA, 2000.