A NEW DIGITAL PURITY?
ON ARCHITECTURES FOR DIGITAL IMMATERIALITY

MARCUS BURKHARDT

The past decade has witnessed the arrival of a new savior – the savior of big data. By means of combining and analyzing unprecedented amounts of data, it promises new modes of knowing the world and knowing ourselves. As early as 1979 Jean-François Lyotard diagnosed the emergence of this new mode of knowledge production, which relies on the resourceful arrangement of data:

"As long as the game is not a game of perfect information, the advantage will be with the player who has knowledge and can obtain information. By definition this is the case with a student in a learning situation. But in games of perfect information, the best performativity cannot consist in obtaining additional information in this way. It comes rather from arranging the data in a new way, which is what constitutes a 'move,' properly speaking. This new arrangement is usually achieved by connecting together series of data that were previously held to be independent. This capacity to articulate what used to be separate can be called imagination." (Lyotard 1984:51)

Even though the many virtues of the knowledge regime of big data that relies both on the radical accumulation of ever more information and its continuous analytical processing can hardly be contested, a critical understanding of the epistemological and ideological underpinnings of the current big data discourse is needed. The following paper is concerned with one of those ideo-epistemological roots upon which big data's promise of salvation is based. My goal is not to debunk the current hype but to ask for the socio-technological conditions of an imagined digital purity, that is, of raw, autonomous and immaterial data, which is at the core of big data and which takes shape as a new data essentialism.

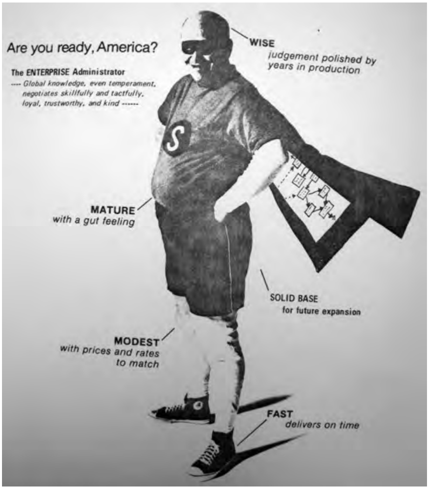

Are you ready, America?
The ENTERPRISE Administrator
---- Global knowledge, even temperament, negotiates skillfully and tactfully, loyal, trustworthy, and kind ------
WISE
judgement polished by years in production
S
MATURE
with a gut feeling
SOLID BASE
for future expansion
MODEST
with prices and rates to match
FAST
delivers on time

Figure 1: The Enterprise Administrator
(Anonymous 1974)

I want to approach this question by starting somewhere and sometime in between. This beginning has no exact date. Yet its location can be exactly specified. The point of departure of my reflections is located in Box 18, Folder 23 of Collection 125 – the *Charles Bachman Papers* – in the Charles Babbage Institute for the History of Computing, Minneapolis (see figure 1). The folder contains a letter sent by the database pioneer Charles Bachman to the *Special Interest List* on *Database Management* on

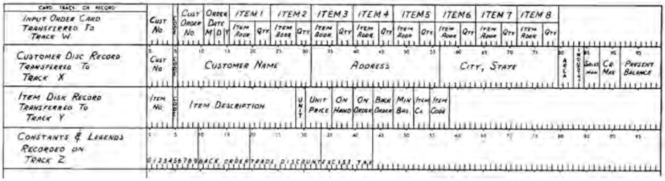
June 25, 1974, informing the List's subscribers on the current efforts undertaken by ANSI/X3/SPARC Study Group on Database Systems. Attached to Bachman's four-page newsletter is a rather odd and somewhat funny image.

On first glance it appears to be an advertisement for database technologies and technicians. However, at second glance it seems to become obvious that it is a parody or – to put it another way – it appears to be an inside joke of the nerdy database community of that time. Since this image is enclosed in the letter without being put into context and with no indication of its source, it became to me one of those strangely fascinating and thought-provoking artifacts that can be found in traditional historical archives. Taking that into account that there is no history to be told based on this picture, but it can serve as starting point for thinking about the logics of database technologies.

The enterprise or database administrator as depicted in this image is one of the superheroes of the digital age. Being 'wise,' 'mature,' 'modest' and 'fast' he is able to supply his customers with a solid and expandable knowledge and information base. Dressed in the typical costume of a superhero, the database administrator hides his identity but saves the day. What is most striking about this depiction is that it reveals the dirty little secret of the administrator's superpowers: on the inside of the cape an information model is drawn. Ultimately, the modeling of information renders it possible that data can be stored in computer databases independently from its future uses in specific applications. This goal had become known in the late 1960s early 1970s as the struggle for data independence.

In 1974 Bachman referred to data independence as one of many "nagging problems" (Bachman 1974:17) in the development of database management systems. The widely recognized objective of separating the management of data from its use in various contexts of application, i.e., from their processing in different application programmes it was a rather abstract goal that needed to be translated in precise engineering problems. As a consequence, data independence turned out to be a fuzzy buzzword for a wide array of different dependencies that ought to be dissolved by powerful database management systems. Here we should name just a few: the physical dependence on specific storage structures and devices, the logical dependence on a specific information model, the dependence on certain integrity and consistency rules, and the redistribution dependence of vast databases which do not just run on a single computer, but on a number of independently operating computers (cf. Codd 1990:345ff.).

The underlying motive of the struggle for data independence was to protect the "investment in data & programs in a changing business & computing environment" (Jardine 1973:2). In other words: information needs are not static but rather change over time and in different contexts of use. The same holds true for the hard- and software database technologies relied upon. These ever changing requirements, combined with rapidly evolving technologies, posed an enormous challenge since application programmes were and in many cases still are dependent on the way in which the required information is stored in computers. Against the background of today's digital media culture, it is somewhat difficult to put oneself in the position of early day database developers and to understand the basic problems they faced.



Card, Track or Record																				
Input Order Card Transferred To Track W	Cust No.	Code	Cust Order No.	Order Date M D Y	Item 1 Item Addr	Qty.	Item 2 Item Addr	Qty.	Item 3 Item Addr.	Qty.	Item 4 Item Addr.	Qty.	Item 5 Item Addr	Qty.	Item 6 Item Addr	Qty.	Item 7 Item Addr	Qty.	Item 8 Item Addr	Qty.
Customer Disc Record Transferred To Track X	Cust No	Code	Customer Name	Address	City, State	Area	Industry	Salesman	Cr. Max	Present Balance										
Item Disk Record Transferred To Track Y	Item No.	Code	Item Description	Unit	Unit Price	On Hand	On Order	Back Order	Min Bal.	Item Cl.	Item Code									
Constants & Legends Recorded on Track Z	0123456789	Back Order	Trade Discount	Excise Tax																

Figure 2: Exemplary plan for the allocation of data on a hard disk storage track (IBM 1957:17)

As an example, think about hard disks as means of storing information. Today, with elaborated file systems or database applications, end users do not have to worry about where their data is physically stored on a hard drive. But when IBM introduced this storage technology in 1956 its users had to know the exact location of data on the 350 Disk Storage Unit, which was part of the 305 RAMAC system, in order to be able to access the desired data. For this reason it was recommended by IBM to their customers to plan the use of the storage allocation beforehand on paper as depicted in Figure 2. Against the background of digital storage technologies, the collection, management, and retrieval of large amounts of information in digital databases take shape as an addressing problem. The seemingly simple question that needed to be answered by database developers was where to put the data automatically and how to retrieve it again. Bachman was faced with this problem in 1962 while developing the Integrated Data Store – in short: IDS – which is commonly considered as one of the first database management systems. The solution to the addressing problem he proposed brings us back to the superhero and his secret weapon:

"This benefit is gained through the structuring of the information itself to permit both associative and multi list referencing of records. This is the means by which the mass memory's ability to retrieve any specified record is translated into the ability to retrieve exactly the information needed to solve a problem. […] The problem given to the IDS is knowing from which pigeonhole to retrieve the required record" (Bachman 1962:IIB-4-3)

Yet this secret weapon hidden in the super hero's cape seems to be a secret lacking secrecy. It is well known that computers have a hard time understanding the data they process. Inside the computer everything is encoded as binary data. It boils down to ones and zeroes, which represent character strings with no obvious meaning to computers. That "John Doe" is a name or that the string "19991231" refers to the date December 31st, 1999 has to be made explicit to computers by describing data with metadata. The structuring of data according to an information model is a common means of making implicit meanings explicit to computers. But structure alone does not suffice for solving the addressing problem in the context of digital databases. The descriptive logic of placing information in the structure of an information model needs to be accompanied by effective procedures for storing, retrieving, updating and deleting information in a database (cf. Bachman 1966:225). This procedural logic determines how information can be handled within computers and how it is put to practice in our emerging database culture. Even though the importance of this cannot be overstated, this paper is concerned with a different question. It aims to show how the gradual solution of the problem of data independence led in recent years to the emergence of a new data essentialism, which resurrects the "transcendental signified" that Derrida (2001:354) among many others put to rest since the 1960s.

Today the transcendental signified takes the shape of the database, which serves as "privileged *reference*" (ibid.:361) interrupting the otherwise infinite "play of signification" (ibid.:354). And the data contained in the "buckets full of facts" (cf. Haigh 2006:33f.) called databases appears to be pure, raw, and autonomous. Of course "raw data is an oxymoron" as Geoffrey Bowker (2005:184) famously stated, but the imagination of digital purity prevails in the recent hype around big data.[1] Big data, however, is just one recent example for this imaginary, whose origins in the context of digital technologies can be traced back to the early years of database development. The subsequent media historical observations aim to underpin this claim by focusing on the debates over how specific information models have to become operative within database systems in order to solve the addressing problem and to ensure the independence of data management from its processing in particular application programmes. This question leads to the efforts undertaken by the Data Base Task Group affiliated to the Conference on Data Systems Languages[2] – in short CODASYL – and by the Study Group on Database Systems initiated by the Standards Planning and Requirements Committee of the American National Standards Institute – in short ANSI/SPARC – to develop an architecture of database management systems.

In 1969 the CODASYL Data Base Task Group advanced the proposition that database management does not rely on just one information model, but on two separate levels of modeling information labeled schema and sub-schema. Thereby, two ways of looking at information were distinguished. The schema describes the way information is stored in the database, whereas the sub-schema defines the way in which the database appears to a specific user group or application programme: "The concept of separate schema and sub-schema allows the separation of the description of the entire database from the description of portions of the database known to individual programs" (CODASYL Data Base Task Group 1969:II-5). Within this architectural framework a database has one schema, but for each schema multiple sub-schemas can be defined which have to be compatible with the database schema (see figure 3). The differentiation of the two levels reflects the competing needs and expectations of different interest groups within CODASYL as William T. Olle stated in 1978 in a retrospective:

> *"The arguments which were raging during the years 1967 and 1968 reflected the two principle types of background from which contributors to the data base field came. People like Bachman, Dodd and Simmons epitomize the manufacturing environment […]. Others, such as those who had spoken at the early 1963 SDC symposium, and indeed myself had seen the need for easy to use retrieval languages which would enable easy access to data by non-programmers."* (Olle 1978:3)

This dispute between engineers and end-users led to the proposal of the two-level database architecture, which must be recognized as a meta-model of information modeling in digital databases. Whereas in the definition of the schema's and the sub-schema's different views of the same information are made explicit, the differentiation between these levels serves as a model of the information flow between the user interfaces and the storage devices. That is, according to this model users do not directly interact with the data storage, but interface with the database through various application programmes by relying on sub-schemas that are mapped on the database schema, which remains hidden from the user.

With regard to the problem of data independence, the CODASYL database architecture quickly proved to be insufficient because in the definition of the schema the conceptual description of information is superimposed by their material organization in the storage: "The schema describes the database in terms of the characteristics of the data as it appears in secondary storage and the implicit and explicit relationship between data elements" (CODASYL Data Base Task Group 1969:2-2).

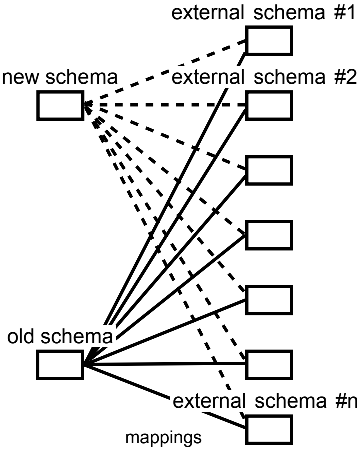


Figure 3: Schema Mapping in the CODASYL Two-Level Architecture (Bachman 1975:570)

As a consequence each change in the ordering of information on hard drives was in fact a change in the schema that again made it necessary to realign the mappings between the schema and its sub-schemas.[3] Shortly after the CODASYL Data Base Task Group presented its final report in 1971 the Standards Planning and Requirements Committee of the American National Standards Institute – in short ANSI/SPARC – founded the Study Group on Database Systems whose task was to determine possible areas of standardization in the field of database technologies (cf. Bachman 1974:16).

Building on the results of the CODASYL task group, a three-level database architecture was developed, distinguishing between the external, the internal and the conceptual views of information stored in databases. According to Bachman the external view or schema is equivalent to the sub-schema of the CODASYL proposal and the internal view as well as the conceptual view are related to the schema. Accordingly the ANSI/SPARC Study Group proposed a more differentiated view on how information is stored in the computer and how its meaning is made explicit to the machine. Whereas in the schema of the CODSAYL architecture the semantics of information was enmeshed in the "layout of physical records" (National Institute of Standards and Technology 1993:54), the ANSI/SPARC architecture proposed the separation of the conceptual description of information from its physical arrangement in storage. The semantic structure and the storage structure of a database are considered to be different levels of looking at and dealing with information. As a result the direct mapping between a schema and its sub-schemas is transformed into a two-step process of translating between the internal logic of computers and the external logic of human users, respectively, between the logics of data management and data processing. In the opinion of the members of the study group this constitutes a certain indirection that is "essential to data independence" (Tsichritzis/Klug 1978:184).

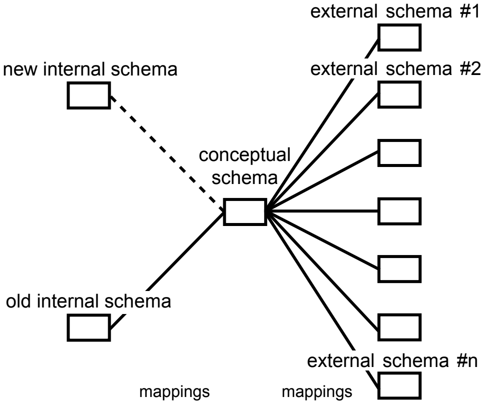


Figure 4: Schema Mapping in the ANSI/SPARC Three-Level Architecture (Bachman 1975:570)

To date this architecture serves as a conceptual but idealized framework of thinking about and designing databases. Almost every basic textbook on database technologies starts by outlining this architecture. Yet in contrast to the original visualization in Figure 4, today the diagram is typically rotated by 90 degrees, thereby emphasizing the flow of information between the surfaces of multiple user interfaces and the invisible depth of the database (see Figure 5). The lasting significance of this architecture is mainly due to the fact that within this meta-model of information modeling the exact level is identified on which the structural explication of the information model becomes operative. The secret weapon of database administrators is the conceptual schema, which is situated in between and serves as mediator between the internal logic of the binary data representation in the storage on the one hand and of the external human uses of information on the other hand. Hereby the external and internal logics of handling vast collections of information are insulated from each other.

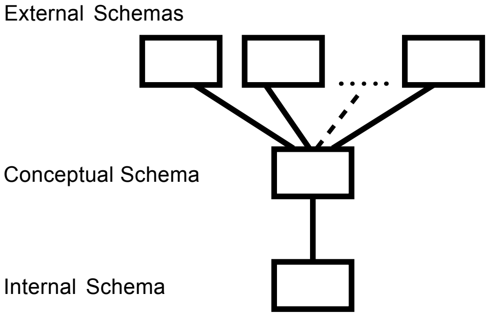


Figure 5: Typical Visualization of the ANSI/SPARC Three Level Architecture

The conceptual schema allows for the automatic storage and retrieval of information in databases because it serves as translator or intermediary. Or, to put it another way, in order to serve as a powerful means for the management of digital information, the conceptual information model has to operate in between and at best must enable the automatic translation of queries submitted by users into effective retrieval routines that can be executed by computers. This is done by database management systems which are usually designed to describe and handle information according to a specific data model. For approximately 30 years the relational data model and, accordingly, the SQL data definition and manipulation language have been predominant.[4] Even though the notion of database systems seems to be equivalent to relational systems in today's digital media culture, the relational modeling paradigm unfolds its efficacy and importance on the basis of ANSI/SPARC three level database architecture.

Within this framework the various external uses of information gain a certain degree of autonomy from the internal management of binary data and their physical materialization in the storage. As a result the end-user of a database can largely ignore the specifics of data management on the internal level. He or she interacts with the database through the information model. In doing so, information is not addressed by location but by its meaning as it is specified in the conceptual schema. This leads to the impression of immateriality accompanying digital information. In this respect Database management systems in general and the ANSI/SPARC database architecture in particular constitute the material basis for the apparent immateriality of the information stored in databases. However, this is not entirely unproblematic inasmuch as this materialized immateriality is accompanied by the illusion that pure and raw information is stored in databases that can be uniformly processed by generic software applications. By shielding users from the internal logic of data storage, many database applications also hide the information's having to be structured in order to become collectable and retrievable. And even if users are aware of this, the role of information models is frequently misconstrued.

On the external level of user interfaces, the database manifests itself as an invisible and inscrutable bucket or container that not only bears a wide array of information but also drives the imagination of its users. This is reflected in the icon conventionally used to depict databases: a barrel or bucket (see figure 6).[5] It is indeed impossible to enter this bucket, that is, and take a look around in the database. In short, we cannot orient ourselves within the database, because users are structurally kept out. The only possibility to explore whether a database has certain information in store is to pose a query that yields an automatic answer. Herein lie the magic and the mystery of digital databases, because the two-step translation process between the external and the internal level takes shape as the direct interaction of users with an apparently inexhaustible resource.

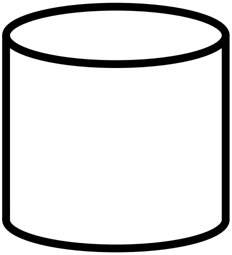


Figure 6: The barrel as iconic representation of databases

As a black box full of information, the database becomes the virtually infinite center of our signifying practices that appears to be "semiotically transcendental" as Alan Liu (2008:217) pointed out in reference to Jacques Derrida. In his essay *Structure, Sign and Play in the Discourse of the Human Sciences* Derrida diagnosed a rupture in the thinking of the "structurality of structure" (Derrida 2001:352) that according to him lead to the "abandonment of all references to a *center,* to a *subject,* to a privileged *reference,* to an absolute origin" (ibid.:361). Yet, within digital database systems the invisible and inscrutable database storage serves as center from which virtually all information can be drawn. It serves as "*a center which* arrests and grounds the play of substitutions" (ibid.: 365), that is, the database delimits the otherwise infinite "play of signification" (ibid.:354) and thus becomes what Derrida called the "transcendental signified" (ibid.:354).

The illusion of the presumed fulfillment of the desire for a privileged reference, center or origin forms the basis of traditional database management systems and their contemporary successors. The Linked Open Data movement, for example, tries to transform the Web into a "single global database" (Heath/Bizer 2011:107) that can be queried and analyzed by "generic applications that operate over the complete data space" (ibid.:5). This promise is based upon the assumption that digital databases enable us to store and retrieve pure information that in turn is evoked by the independence, autonomy or immateriality inherent to digital information within the architectural framework of database systems. In Michel Foucault's terminology of the *Archeology of Knowledge,* pure information could be described as statements without enunciative function, that is, statements whose identity does not rely on "a complex set of material institutions" (Foucault 1972:193). The belief in this new digital purity manifests itself in Tim Berners-Lee's well-known call for "raw data now" (2009) which forms the ideological basis of the semantic web vision and the linked open data movement.

Contrarily, imaginary databases are never just collections of preexisting information. They are rather means of creating information by transforming them into a resource. Or to put it in terms of Heidegger's (1977) philosophy of technology, a database transforms information into a standing-reserve that is ready-to-hand, whereby the information comes into existence as information by means of the conceptual information model which delimits what can be stored within the database. In regard to the information model, the database does not represent reality but constructs it by defining what is to be "counted-as-one" (Badiou 2005: passim) and is therefore to be treated as existent according to that model. Yet within the limits of this model the database might contain objective information about reality. In this regard databases oscillate between social constructivism and realism.

Taking this double *nature* of database information into account prevents us from naturalizing data and treating it as pure, autonomous, and immaterial. Database technologies are rather the material basis for the seeming purity, autonomy, and immateriality of digital data. As such they do not just determine what we know about the world, but what it means to know 'the world,' which will always already have been 'our world.'

Notes

[1] Drawing on Bowker‘s statement, Lisa Gitelman (2013) published an edited volume on the history and theory of the emerging data culture.

[2] Originally the CODASYL Data Base Task Group was founded in 1965 as *List Processing Task Force* aimed at extending the programming language COBOL with capabilities for handling large datasets. The group, which was set up of users and developers from the computer industry, later renamed itself and focused on developing an architectural model for database systems as well as the network data model.

[3] By distinguishing between how information is stored in the computer and how it appears to users the schema-sub-schema-architecture has certain parallels to the common twofold view on digital objects that manifest themselves as binary representation invisible to the human eye and as phenomenal presentation on user interfaces. (vgl. National Institute of Standards and Technology 1993:47)

[4] The relational data model was proposed by Edgar F. Codd in his seminal paper *A Relational Model of Data for Large Shared Data Banks* (1970). During the 1980s the relational model became the de facto standard in database management.

[5] Incidentally, the same pictogram is often used for depicting hard disks in technical contexts.

Bibliography

Badiou, Alain. Being and Event, New York: Continuum, 2005.

Federal Information Processing Standards Publication 184, 1993, URL: http://www.itl.nist.gov/fipspubs/idef1x.doc.

Foucault, Michel. The Archaeology of Knowledge and the Discourse on Language, New York: Pantheon Books, 1972.

Framework: Report of the Study Group on Database Management Systems," *Information Systems,* 3, 1978. p 173–191.

Gitelman, Lisa (Ed.). 'Raw Data' is an Oxymoron, Cambridge: MIT Press, 2013.

Haigh, Thomas. " 'A Veritable Bucket of Facts': Origins of the Data Base Management System," *SIGMOD Record,* 35, 2, 2006. p 33–49.

Heath, Tom, and Bizer, Christian. Linked Data: Evolving the Web into a Global Data Space, Synthesis Lectures on the Semantic Web: Theory and Technology, San Rafael: Morgan & Claypool, 2011.

Heidegger, Martin. "The Question Concerning Technology," *The Question Concerning Technology and other Essays,* New York: Garland, 1977. p 3–35.

IBM. 305 RAMAC Manual of Operation, New York: IBM, 1957.

Jardine, Donald A. "Data Independence: Ad Hoc Presentation" *Charles W. Bachman Papers (CBI 125),* Box 16, Folder 30, Charles Babbage Institute, University of Minnesota, Minneapolis, 1973.

Liu, Alan. Local Transcendence: Essays on Postmodern Historicism and the Database, Chicago: University of Chicago Press, 2008.

Lyotard, Jean-François. The Postmodern Condition: A Report on Knowledge, Manchester: Manchester University Press, 1984.

National Institute of Standards and Technology. Integration Definition For Information Modelling (IDEF1X)

Olle, T. William. The Codasyl approach to data base management, Chichester: Wiley, 1978.

Tsichritzis, Dennis, and Klug, Anthony. "The ANSI/X3/SPARC DBMS

SEARCH ROUTINES: TALES OF DATABASES

24/7 48
Access 75, 91, 94, 113, 124
Accumulation 87, 109
A certain kind of desire 41
Addressing problem 91, 92, 93
Alchemy e-books 72
Algorithm 10, 27, 28, 35, 36, 47, 69, 73
All aspects of interaction 109
Amounts of data 87
Anonymous corporation ...11, 75, 77
Assembly line 54
Church of super-computers 53
Common male phantasy 36
Computing and gender 42
Continuity 114
Data essentialism 88, 92
Deer and data center 32
Embeddedness 109, 118
Extendable shopping card 31
Hard drive 32, 33, 91, 96
Information model 89, 93, 98
Infrastructure 12, 107, 108, 110
In the woods 67, 70
Invisible 55, 64, 97, 101, 122
Just-in-time 42
Love story 35, 39
Magic 12, 67, 100
Management 51, 88, 90, 98
Materialized immateriality 99
Membership 108, 118
Mirrored 32, 48
Narrator's voice 55
Ownership 76, 115
Pigeonhole 92
Post-Fordism 36, 38, 54, 58
Query mode 100, 123
Random access disc 11, 38
Relational database 115, 118
Resistance 114
Shared folder 47
Subconsciousness 42
Sub-schema 93, 94
Surplus labor 57, 58
Technological subjects 42
Things we connect 29
To disappear 64
Universe of boring things 28
User interfaces ... 94, 97, 103, 124
Weather forecast 11

D21 KUNSTRAUM

IMPRINT

Editors: D21 Kunstraum Leipzig; Lena Brüggemann,
Francis Hunger
Copy Edit: William Clapp, Juliane Richter
Proofreading: Philipp Anders
Design: Paul Spehr

Printed by: Books on Demand GmbH,
Norderstedt

Die Deutsche Bibliothek verzeichnet diese Publikation in der deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über http://dnb.ddb.de abrufbar.

The digital edition of this book can be downloaded freely at www.d21-leipzig.de