· Part I ·

Objects

This page intentionally left blank

· CHAPTER 1 ·

The Genesis of Digital Objects

Digital Objects and Their Milieux

We are currently living in a digital milieu; we Facebook, we blog, we Flickr, we YouTube, and we Vimeo. Nouns and brands have become verbs, even forms of life. The speed of technological innovation, the ubiquity of the latest and greatest versions of electronic devices, the promise of an emancipative technology or media, financial investment based on the digitization of human relations, and so on—they all constitute a seeing that is never in the present but is rather the projection of a nihilistic not-yet. This mode of existence is not what Martin Heidegger calls "temporal ecstasy," in which one nonetheless grounds oneself in an authentic time; it is rather a hyper-ecstasy that celebrates speed while simultaneously being haunted by the anxiety of not being there, not being able to situate itself within the grand rhetoric of the technology evangelists. I call this experience *technological ecstasy,* a way of becoming that has no clear idea of its direction yet is characterized by acceleration and adventure. The constant passing of the "new" constitutes an *indifference toward rhythm,* which, in turn, legitimatizes a natural seeing of what is there and what is expected. The word *new* denotes the passing away of the old and the differentiation of the world in its projection, driven by a gigantic force of movement.

The understanding of technology is no longer a matter of a cultural critique of technology. Indeed, the traditional exclusion of technology from culture must be brought into question. To resolve this conflict we must employ a new *organon,* or a new series of philosophical propositions. Any proposed theory would initially need to identify the reality with which it is concerned. To understand the "real," we must compare it with what is commonly understood as virtual. The idea of the virtual, which was popular some years ago as a descriptor of certain kinds of community and interaction dependent on digital media, such as online forums and cybersex, has since receded into the background, as you can no longer say today that someone using Facebook or Second Life is living within a virtual world (considering that he is interacting with his real friends and engaging in activities like providing his credit card number and personal information to order a Swedish Visa online).[1] The introduction and convergence of technologies like Bluetooth, Wi-Fi, and GPS allow for more accurate contextual and geographical detections, leading us into the REAL. How can we address this digital milieu? It is another world, a strange world, one that is simultaneously artificial and natural. It is as complicated as what we used to refer to as the "real world," and more important, it is a world we are already *in*.

Our investigation will focus on digital objects to better understand where the current transformation process is heading and to develop an appropriate method for its investigation. The term *digital object* remains ambiguous here, because the vast quantity of digital objects are comparable in breadth and diversity to the vast array of animal species. Instead of addressing all of them, I will be focusing mainly on data and metadata, which embody the objects with which we are interacting, and with which machines are simultaneously operating. The first questions we will ask at this point are, does hardware count? What about algorithms? Although I am tempted to include all objects related to computation as digital objects, some restriction of scope is necessary to allow me to focus an equal amount of attention on the digital aspect of the digital object. We have a tendency to call everything an object, to generalize all computational components as digital objects. However, this approach appears to be rather problematic, because individual objects would lose their singularities. The same issue applies when object-oriented philosophers give the general name of "objects" to all entities apart from the human being. Thus it is necessary here to *suspend* any common understandings or interpretations of "objects." It is true that we are able to reduce all operations to 0 and 1 binaries, and even further down to the activities of electrons and atoms; however, this only gives us a particular order of reality in terms of what *digital* means, and one that has little to do with the direct experience of the users. Digital, in the context of this book, has a specific orientation toward the automation of data processing. Data directly intervenes throughout our human experiences in a double sense. When we look at the term *data,* we generally do not recognize its Latin origin, as the plural form of *datum,* meaning "[a thing] given." The French word for data, *donnée* ("given," from *donner,* "to give"), retains the Latin sense exactly. If data are the "things" given, then what is it that gives data? Aside from having the speculation in mind that this givenness comes from God, we should recognize that since 1946, the word *data* has had an additional meaning: "transmittable and storable computer information."[2] This second understanding of *data* suggests the need for a reconsideration of the philosophy of objects, because it can no longer be assumed to refer entirely to sense and noetic data. Instead, one should recognize this translation as taking on a material form and consider how this materiality constitutes a new form of "givenness." The significance of the recent development of data processing, that which we have since proclaimed as the digital, demonstrates the extension of data—exchanging capabilities beyond individual computers such that we can process large amounts of data by establishing connections to form data networks that extend from platforms to platforms, and from databases to databases, constituting a technical system.

The next question we face is, how should digital objects be conceptualized? According to the common view of scientists and/or mathematicians, we can have a superset of objects, inside which we can find a subset of objects called technical objects alongside natural objects, as theorized by Gilbert Simondon. It is also understood that within this subset, we can find a further subset of objects called digital objects. It is possible that there may be more subsets than those which have previously been accounted for, according to different schemes of classification. Instead of following this classification, however, I would like to propose a splitting between technical objects and digital objects. Digital objects are new forms of industrial objects. If the "new" demands a new understanding, then addressing this may begin with asking where this "new" came from. The new can only manifest relative to the old, either as a continuation or as a break or rupture. As Simondon would say, inventions always attempt to remove the obstacles and resume a general continuity of development.[3] The analyses throughout this book will be primarily concerned with a series of incompatibilities created by the reverberations of the new, those that demand we direct our attention toward the genesis of objects within a historical perspective. In this chapter, I describe the genesis of digital objects by situating them within the history of computing and introduce the analysis of Gilbert Simondon. I compare the relation between data and objects in the new setting and how this account of their genesis can contribute to our understanding of computational technologies.

The Double Movement of Object and Data

The methods through which objects become translated into data are not new. They follow the logic of digitization after the emergence of the modern computational machines, namely, everything can virtually be represented in digital formats. There are two dominant forms of digitization: the first follows the system of mapping or mimesis (for example, the production of digital images, digital video, etc., which are visually and repetitively distributed throughout the physical world), whereas the second takes place by means of attaching tags to objects and coding them into the digital milieu (by means of this digital extension, the object then obtains an identity with a unique code and/or set of references). The second movement of objectification of data comes a bit later. I call the first process the *objectification of data* and the second process the *dataification of objects.* In saying this, I don't mean to say that these things are not objects before they are objectified by metadata schemes but rather that they are formalized as objects through human agency and then recognized as objects by computers; or, in the spirit of Heidegger, they are things *(Ding)* before they become objects *(Gegenstand).* This way of representing objects is widely known as knowledge representation. *Knowledge representation* has been a key topic within artificial intelligence (AI) for decades, and it is steadily increasing in importance again following the failure of a number of large-scale projects[4] now under the name "semantic web." This objectification process has two very significant implications: (1) it breaks away from the *hyperlink-based* Web to become the *object-based* Web and (2) it signifies a more significant role for the machine, not as an input–output device, but also as a partially "thinking machine." I want to approach this development in terms of two technical questions, which are simultaneously philosophical questions: the question of objectification and the question of intentionality and experience pertaining to thinking machines. Indeed, this book is the result of an endeavor to read the history of philosophy through digital objects and at the same time to read the history of digital objects through philosophy. Finally, we will see that computation is no less philosophical than philosophy, and philosophy is no less technological. To pursue this path, we need to unfold the technical details of the emergence of digital objects before proceeding to a more philosophically oriented analysis.

My reading of the movement of the Web sees it as the inauguration of a process of the objectification of data, not only for humans but also for machines. It is in this sense that the founder of the Web, Tim Berners-Lee, could envisage the emergence of a "global mind" shared between humans and machines and supported by the Web.[5] In 1989, when he proposed the World Wide Web at CERN (the Swiss-based high-energy physics lab), his model was largely influenced by the technology visionary Ted Nelson, although with some fundamental differences. For Nelson, the concept of a digital object was impossible, as he saw the network from the point of view of literature. Nelson's idea of hypertext was to realize nonsequential writing[6] through which the interconnectivity of literature can be unfolded in different temporalities. Every hypertext would imply a jump from one spatiotemporal setting to another, while through these trajectories, a network can be understood as a form of nonsequential writing.

Nelson's vision was restricted by its dependence on the limited concept of text and writing, whereas Berners-Lee's focus on the Web in the 1990s was primarily concerned with hypertext and the hyperlink. The striking difference between Berners-Lee's model and Nelson's model reflects their fundamental motivations. Nelson's vision of the Web was tied to a payment system, so that the payment to the authors of the literature could be managed by links. This motivation coincidentally led to a completely different architecture of links from that of Berners-Lee's model. Nelson proposed, in his Xanadu project, a two-way link system, while we know that the early Web was a one-way-link-based system on <a href>, which specifies the URL to be loaded when the link is clicked. Today these two-way links have been realized, not as Web architecture, but as overlays, such as blog comments, trackbacks, and so on. Berners-Lee's vision comes from the internal sharing of documents within CERN, so that different versions of documents could be linked and archived in a way that would minimize the loss of information in a "final report." Nelson was to some extent justified when he criticized the Web as a file system with one-way links: "today's one-way hypertext—the World Wide Web—is far too shallow. The Xanadu project foresaw world-wide hypertext and has always endeavored to create a much deeper system. The Web, however, took over with a very shallow structure."[7] But it is not an entirely fair comment, because we must also understand that for Berners-Lee, the Web in its evolution has already far surpassed this stage of file sharing.

For the Berners-Lee of the 2000s, the vision of the Web has already developed beyond the sharing of documents to the collaborative imagination of minds and machines. This is more or less based on the assumption that the mind perceives objects through representations. Structured metadata provide the computer program with the conceptualization of objects. The formal definition of metadata is "data about data." An intuitive example is the library search: when a person looks for a book in the library catalog, she must submit different information, for example, the name of the author, the title of the book, or the ISBN number. This information, which is in addition to the content itself (data), is known as metadata. The formats within which these data are presented are called *metadata schemes.* We can compare this with Kant's schemata, as the fusion of the pure concepts or categories that gives rise to phenomena from sense—data. In the age of hypertext, online objects are only meaningful to humans, not to machines. However, in the age of metadata, online objects are considered to be meaningful to both machines and humans.[8] Machines understand the semantic meaning of objects via the structures given to the metadata. This objectification movement is called the semantic web, introduced by Tim Berners-Lee in 2001. Berners-Lee argued that "in the future, when the metadata languages and engines are more developed, it should also form a strong basis for a web of machine understandable information about anything: about the people, things, concepts and ideas."[9]

The double movement from object to data, and from data to object, will be an ongoing project that will continue to develop over the coming decades. It presents us with new forms of objects, constituting a new milieu in need of further reflection. This is the case not only within the Web industry but also throughout information science as a whole. If we reflect on the early stages of the development of the catalog system within library science, we can see that it followed the same technological tendencies. The Web (or simply the Internet in general) promotes a milieu that includes various sectors influenced by a combination of technological, economic, and political concerns. For example, in library science, early cataloging schemes like Machine Readable Cataloging (MARC) and Angelo-America Cataloging Rules (AACR) grounded a lengthy effort to address the question of annotation. However, since digitalization and Internetization, these schemes have become obsolete and are being replaced by ontologies, such as Dublin Core (DC).[10] The reason for this is twofold: first, MARC and AACR are specific protocols that cannot be used outside of their limited field, implying that they cannot effectively be integrated into the digital milieu alongside other machines. The second reason is that they cannot be read by humans and are thus unable to

```
245 10 Rhkjsow fijkslw bf ksjk jsiousol/$c w Hfuyse can Lqzx
250 2c pj.
260 0 Klana:$b Fry Psgh, $c 2001.
300 232p.; $c 28 cm.
```

Figure 7. Information for a book in the MARC format.

participate in the universal communication of the "global mind." In other words, they do not treat a book as an object as such but rather as mere symbolic data. Figure 7, an example of MARC, provides the informational data for a given book.

"MARC must die"[11] is a familiar slogan commonly expressed among library technicians since the early 2000s (a website created by digital librarians is dedicated specifically to this cause). This has also presented a crisis regarding the creation of digital objects. Because librarians and technicians working with digital objects must manage a great magnitude of symbols that provide them with hardly any concrete or understandable information, they are condemned to be the assistants of machines. This has led to one of the most notable phenomena of alienation within the digital milieu. The vision of the semantic web, as a means of generating new forms of objects that are meaningful to both humans and machines, received a lot of appreciation and interest from various communities. The genesis of digital objects is hence not the sole effort of Tim Berners-Lee and his team in the World Wide Web Consortium but rather a milestone achieved via the advancement and development of the computation as a result of its long history.

Throughout the following sections, we examine the historical emergence of digital objects by attending to the thoughts and arguments of Gilbert Simondon. The importance of introducing Simondon's thought here in this context is that he was not only perhaps the first thinker to propose a philosophical understanding of technical objects but he also conceived the development of a technological culture to serve as a solution to Marx's critique of alienation. At the very beginning of *Du mode d'existence des objets techniques,* Simondon wrote, "The stronger cause of alienation in the contemporary world resides in this misunderstanding *(méconnaissance)* of machine, which is not an alienation caused by machines, but the lack of understanding *(non-connaissance)* of its nature and its essence, because of its absence from the world of significations and its omission in the table of values and concepts belonging to culture."[12] Simondon introduced a potential approach called *mechanology,* which would put machines at the forefront of general education, proposing that technological knowledge should be introduced as part of the educational curriculum, with a status equivalent to that of literature.[13] Throughout the history of metaphysics (which in Heideggerian terms equates to the history of philosophy), from as early as Plato to the later theories of Edmund Husserl, a technical object was nothing more than a tree in the garden or an apple on the table. What interested philosophers was either the idea of and the essence of the objects, as manifest in Plato's *idea,* Aristotle's *form and matter,* Descartes's *extension,* Leibniz's *monads,* Kant's *schematization,* Hegel's *dialectics of consciousness,* or Husserl's *noetic and noematic* correlation, or a natural (or organic) and mechanical opposition was posed, which consequently subordinates the mechanical to the natural. Technological knowledge simply did not achieve a formal position within this philosophical tradition, except at the brief moment of Diderot and D'Alembert's encyclopedia, which attempted to render technical knowledge transparent to the public. The emergence of cybernetics in the early twentieth century, however, stimulated a rupturing of the philosophical tradition by questioning the border between the natural and the artificial. The dynamic of machines cannot be captured solely by an *eidos.* This coincidentally created a demand for a new direction in philosophical thinking during the mid-twentieth century, from which emerged the two very contrasting approaches that are of particular interest to us here. On one hand, Martin Heidegger lamented that cybernetics marked the overall completion of metaphysics and simultaneously the end of philosophy. This caused him to attempt a retrieval qua retreat to a new form of thinking. On the other hand, Gilbert Simondon wanted to understand technology not as a closure but rather as a process working toward the perfection of technical individuals and through a systematic understanding of the transformation of the human with the evolution of tools to search for a technical disalienation.

Individualization of Technical Objects

We should first address two prominent concepts used by Simondon that are often confusing for his readers: individuation and individualization. For Simondon, individuation is clearly different from individualization. Individualization concerns functions such as somatic specializations and psychic schematization. When the term is applied to living beings, it denotes the development and division between the psychic and the soma. Individuation, alternatively, concerns the genesis and resolution of tensions to arrive at a metastable equilibrium passing by a restructuralization of relations.[14] Individualization is not opposed at all to individuation; they would be better viewed as two separate orders of magnitude of beings. In *L'individuation à la lumière des notions de forme et d'information* (2005), Simondon talked about the individuation of physical beings (e.g., crystals), living beings, and the individuation of psychic beings; in *Du mode d'existence des objets techniques,* he mainly talked about "technical individualization" rather than "technical individuation." Could we also discuss the "individuation of the digital object"? Simondon's hesitations toward this leave us with a rather large space of inquiry, allowing us to develop his efforts further. To expose these possibilities, it will be necessary for us to observe and analyze how Simondon carried out his analysis of technical objects.

A technical object is always a product of determination, or even overdetermination. The term *overdetermination* refers to a process of imposing constraints and conditions so as to maturate the functionalities of the technical objects. The maturity of technical objects can then be measured by what Simondon calls *technicity,* which is the degree of concretization within the object. Simondon sees the evolution of technical objects as a progression from abstract objects to concrete objects. To be concrete entails the convergence and adaptation of the object to itself. For example, when a technical object integrates further functions into itself and subsequently compromises these functions in a coherent way, it becomes more concrete than it was previously; as Simondon wrote, "the unity of technical object, its individuality, its specificity, are the characters of consistence and convergence of its genesis."[15] Hence we can say that industrial technical objects are more concrete than the artisan's products. Simondon argued that the customized products belonging to the artisans are not technically essential but that rather they are produced by other essential factors, such as external needs—whereas in industry, technical objects gain their own coherence. Simondon's technical objects are therefore also industrial objects.

According to Simondon's classification, there are two forms of technical object, namely, "element" (or "infra-individual") and "technical individual."[16] In comparison with the elements that are simply building blocks, the technical individual has a complete set of functions as well as a mechanism that allows it to maintain the internal stability in response to specific external disturbances. Simondon defines a technical individual as "one having an associated milieu as a *sine qua non* condition of its functioning." The associated milieu is a means of adaptation, ensuring that the individual is "not to be influenced by the external technical and natural environment."[17] This criterion implies that the object already has the ability to independently stand on its own within the constraints that are already set into its overdetermination.[18] Technical individualization for Simondon depends on the discovery and invention of its associated milieux:

> The principle of individualization of technical objects in an ensemble is therefore the principle of the sub-ensembles of recurrent causality in the associated milieu; all the technical objects that have a recurrent causality should be separated one from the others and connected in a way to maintain this independence of their associated milieux.[19]

We should note here that it is necessary to keep the associated milieux separated; otherwise, the unified associated milieu would become the Achilles' heel. Simondon's technical individual in this instance specifically refers to a hardware system rather than to digital objects, which consist mainly of code. At first glance, we cannot reuse Simondon's vocabulary to understand digital objects, because there is no such reciprocal causal mechanism inside the digital object that allows for its self-stabilization.[20] Alternatively, however, we can see that databases, algorithms, and network protocols become the associated milieux of digital objects. And as a digital object is also a set of logical statements, its reciprocal causality is highly controllable. The associated milieu cannot be thought of only as a mechanism inside the individual but should instead be considered as something in between the exterior and interior milieux. When Simondon discusses nonindustrial civilization as a time when humans do not have industrial technical individuals (because they only use simple tools), he says that man's "apprenticeship leads him to technical self-individualization. He becomes the associated milieu of the different tools he uses."[21] Humans created the associated milieu for the tools through their gestures and habits, stabilizing and regulating the entire ensemble: tool—bearers themselves became technical individuals.

In this sense, we are able to identify the associated milieux for digital objects, each of which is further stabilized by the specific network in which it is situated, additionally including its users, data structure, network protocols, and so on. To be stabilized by the system, it must also include various mechanisms that regulate it. The evolution and concretization of these mechanisms allow a digital object to develop and integrate an associated milieu of its own, which is what Simondon calls a technical *individualization,* whereby something corresponds to what was illustrated before as the "objectification of data" or schematization. This process of individualization consists of three parts. First is the synthesis of data through the metadata scheme, which is comparable to Kant's concept of the apprehension of objects. Second are the built-in constraints within the object, giving digital objects the capacity to regulate their identity within the digital milieu. For example, when considering an ontology of kinship, one can only have one mother and one father. And third, the object has now become a logical entity, hence it expresses a logical infrastructure as a constituent of the digital milieu. I will further demonstrate these three stages of the process in the following sections of this chapter. To push this still further, a digital object is also constantly in the process of reestablishing and renegotiating its relations with other objects, systems, and users within their associated milieux. Digital objects also take up the functions of maintaining emotions, atmospheres, collectivities, memories, and so on. This gives us a dynamic and energetic understanding of digital objects. I want to distinguish this process as *individuation.*

As part of an industrialized civilization, human beings have begun to lose their role as technical individuals, as they become mere operators, either pushing a button, moving raw material, or cleaning the machine. This does not necessarily mean that the human's position in the associated milieu will become any less important than it already is, or that humans will inevitably be ejected from the milieu as a whole. It is rather more likely that they will slowly become deskilled, and their technical knowledge, which indicates their affinity to machines, will be reduced to the most superficial level. This, for Simondon, is the problem of alienation raised by Marx. Simondon compared the relation between technical objects and the human as the relation between the musician and the conductor, as each produces an affect and is mutually affected by the other.[22] As with technical alienation, however, this mutual relation is destroyed. For Simondon, restoring this mutual relationship would be a means for developing a technological culture. Does the current technological transformation offer us the possibility of doing this? On a social networking website, digital objects are not able to function on their own without the activities of human beings, who create and modify them. Without this intervening creation and modification, machines would have nothing to process. The new demands that are placed on humans do not mean that they regain their importance, however. As we will soon see, a change occurs within nature regarding man's existence and experiences within technical systems. On one hand, we are witnessing humans becoming nothing more than digital objects themselves. However, on the other hand, we may also appreciate that they are integrating with machines, inaugurating a new set of operations under the names of *social computing* and *crowd sourcing.* We now have two fundamental understandings: first, that technical individuals individualize (Part I) by adopting and creating an associated milieu to stand independently, and second, that individuals individuate themselves through the collective—an assemblage or a network of relations and associations in its world (Parts II and III). To go into this further, we will need to address the concretization of digital objects.

From GML to HTML: Form as Technical Tendency

The development of technicality is a process motivated by various interruptions and discontinuities. New technologies are able to cut through the lineage, giving it new directions. These directions may collide and diversify its progress into different paths; however, these diversities will be synchronized by a dominant technical tendency. The French paleontologist and paleoanthropologist André Leroi-Gourhan distinguishes technical tendency from technical fact. The former is universal and abstract, whereas the latter is particular and concrete, closely related to its milieux, that is, geography, ethnicity, climate, and so on. We can further distinguish different degrees of fact according to different modes of adaptation within ethnic groups. Technical tendency is inevitable and foreseeable; technical fact is unforeseeable and requires certain local inventions rather than direct borrowing from other groups.[23] Leroi-Gourhan gave the example of forging. There are no technical tendencies of forging, only technical facts that depend on a varied range of conditions such as fire, metal, combustion, fusion, commerce, mode, or religion. The technical tendency is the force that traverses the various milieux and cultural differences, for example, the universal invention of wheels as a means of carrying heavy loads and handles for flint.[24]

The separation of form and matter, as evident among technical inventions, is a technical tendency in this sense. Digital objects follow such a tendency. The semantic web is a specific technology utilized in computation among many users. It subsequently deviates from IBM's Generalized Markup Language (GML) and from knowledge representation in AI (while incorporating some of their core concerns). Simondon called this process "the time of relaxation," which equates to "the real technical time. It can become more dominant than all other aspects of historical time, to the extent that it can synchronize all other rhythms of development and appear to determine the whole technical evolution, whereas in fact it merely synchronizes and induces evolution phases."[25] Synchronization means convergence, which also demands a new form of technicity. This technical time is also the time of the technical perfection of objects, regarded as "a practical quality or, at the very least, the material and structural support for certain practical qualities."[26]

GML was invented in the late 1960s by IBM, at a time when the Web hadn't yet come into being. It served as a solution to a project that would require the integration of a text editing application with an information retrieval system and a page composition program. These applications could not be run on the same machine until Charles Goldfarb and his colleagues invented GML in 1969, a markup language that standardized the structure of the document:

> This analysis of the markup process suggests that it should be possible to design a generalized markup language so that markup would be useful for more than one application or computer system. Such a language would restrict markup within the document to identification of the document's structure and other attributes. This could be done, for example, with mnemonic "tags." . . . The actual processing commands, however, would not be included in the text, since these could vary from one application to another, and from one processing system to another.[27]

GML consists of application documentation, which defines the data according to tags, and Document Type Definitions (DTDs), which subsequently define these tags. We can draw two conclusions here: (1) the markup language gives "semantic" meaning to the data through the distinction of tags, so that the application will be able to process the data as an object and parse useful information, which will result in the first step of data organization, and (2) the markup language provides a solution to the problem of the incompatibility of applications and machines; in other words, it can connect all machines by presenting them with a common protocol. The concept of universality is very important in the history of the development of the Web, which, as conceived by Berners-Lee, is a universal space.[28] GML separates the content from the form (metadata scheme) by acquiring the knowledge of the form, whereby the machines are not required to understand the semantic meaning of the entire content. This universal space is also determined by the universalization of the forms involved. These can be in the form of metadata schemes, protocols, or any other standard forms. This form versus content–matter hylomorphism has been a key concept in traditional metaphysics since the time of Plato and Aristotle. Matter subsumes itself as forms to actualize itself. Form is also a way of accessing the universal, because it provides idealities and particularities.

In 1986, the International Standard Organization (ISO) adopted an advanced version of the GML—later known as SGML, or Standard Generalized Markup Language—which prepared the pathway for the establishment of HyperText Markup Language (HTML) in 1991.[29] HTML is a subset of SGML, but with a fixed DTD. The motivation behind HTML, following SGML, was strategical and partly political, as SGML was the dominant protocol at that time, and HTML can hence be more easily accepted by the community. Nevertheless, its separating of content and form was also a step with technological significance. Berners-Lee wrote that "an architectural rule which the SGML community embraced is the separation of form and content. It is an essential part of Web architecture, making possible the independence of the device mentioned above, and greatly aiding the processing and analysis."[30]

Hylomorphism and Individualization

Here we should first place the concept of *hylomorphism* in its correct critical position. It is the most intuitive idea about technology, as suggested by Aristotle when he stated, "In speaking here of matter I have in mind, say, the bronze of a statue, while by shape-form I mean the geometry of the object's appearance and by the composite the statue itself as a whole entity."[31] One can make the criticism, as both Simondon and Heidegger did, that matter is not the passive object of the form but rather that form derives from matter. A good artisan creates the statue based on a particular status of the material or in seeing the form arise out of matter.[32] This critique is nevertheless based on human experience, and it was valid specifically within the age of artisanal production. In the age of mass production, however, this superiority of matter over form is reversed, because it is no longer a question of human skill but rather of the machine standards that create such forms. Form and matter here have two contrasting meanings: (1) form is the compensation for the machine's inability to understand the semantic meaning of the content (comparable to the metaphor of molding, which is always a standard), and (2) form activates a pursuit of ideality that becomes a point of convergence for Western metaphysics with modern science and technology, or what Martin Heidegger would call the onto-theological constitution of things. The conceptualization of form over matter in the age of machine production exposes an innate contradiction within modernity. On one hand, the production process has sped up significantly due to the homogenous mold, which largely ignores the singularity of matters. On the other hand, form replaces all situational discourses with a set of rigid rules that further constitute various forms of life externally. This double-bladed argument continues to fuel an ongoing social debate, and yet a radical interpretation of form is still lacking.[33]

The architect Christopher Alexander, in his book *Note on the Synthesis of Form,* writes that "the ultimate object of design is form. The reason that iron filings placed in a magnetic field exhibit a pattern—or have form, as we say—is that the field they are in is not homogeneous. If the world were totally regular and homogeneous, there would be no forces, and no forms. Everything would be amorphous. But an irregular world tries to compensate for its own irregularities by fitting itself to them, and thereby takes on form."[34] For Alexander, a design problem can only be solved by form, and the content of the problem is defined by its context. This somewhat resonates with what we have seen in the introduction of the computationism of Chaitin and Fredkin. It is therefore necessary to distinguish the form as a technical tendency from the perception of technical objects in terms of their forms. However, in contrast to this conception of form as the ultimate force of production, Simondon suggests that a tool "is not made of matter and form only. It is made up of technical elements arranged from a certain system of usage and assembled into a stable structure by the manufacturing process."[35] Despite the fact we know that mass production is mainly based on molding and the form–matter logic inscribed in it, the technical process cannot be simply explained by the principle of hylomorphism. The identity of a technical object equates to the totality of its production, as opposed to its form and matter. Simondon puts this in a rather extreme way: "There would be no exaggeration in saying that the quality of a simple needle expresses the degree of perfection of a nation's industry."[36] This marks the departure from the individual determined by form toward a broader discourse of systematic determination. Indeed, both processes point to what Simondon calls the "historical singularity": production itself is always the product of a historical moment distributed throughout the entire technical ensemble. Simondon suggests that despite hylomorphism being insufficient to account for the current nature of technological production, it is still nevertheless an intuitive mode of thought that remains a dominant engineering principle. My hypothesis is that under different historical and technical conditions, hylomorphism produces something other than its intended effects in material terms. It consequently exposes the limits of the thinking that reproduces itself; hence our analysis must first place form under suspicion and reposition it as our analysis unfolds.

HTML was implemented for the World Wide Web in 1991 and has remained the standard language that we use today. During the early days of the HTML markup scheme, metadata mainly focused on the structural, visual, and hypertextual representations of the page. The formalization and limitation of vocabularies has reduced its complexity, producing a light and portable language. In comparison with the Java programming language and the Web-based Java applet, HTML is very limited in terms of its programming power. Berners-Lee calls this approach based on simplification the *principle of least power.*[37]

A metadata scheme, as a relatively weak language, expresses only forms, instead of having the capacity to manipulate forms and objects, which is what occurs within the Java programming language. HTML uses a set of standardized tags to indicate content representation in a logical format. As in the simple example of HTML in Figure 8, <p></p> denotes the inclusion of a paragraph (as structural), denotes a bold font (as visual), and denotes a hyperlink (as hypertextual). We can probably say that HTML is a metadata scheme. As a fairly weak or

```
<!DOCTYPE html>
<html>
        <head>
        <title>Hello World</title>
        <head>
        <body>
            <p><b>Hello World!</b></p>
            <p><a href = http://helloworld.org>hello world</a></p>
        </body>
</html>
```

Figure 8. A simple example of HTML.

The IMG element allows another document to be inserted inline. The document is normally an icon or small graphic. This element is *not* intended for embedding other HTML text.

SRC	The value of this attribute is the URL of the document to be embedded. Its syntax is the same as that of the HREF attribute of the A tag. SRC is mandatory.
ALIGN	Take values TOP or MIDDLE or BOTTOM, defining whether the tops or middles or bottoms of the graphics and text should be aligned vertically.
ALT	Optional text as an alternative to the graphics for display in text-only environments.

Figure 9. Specification of an image in early HTML protocol.

ineffective programming language, it does not provide the machine much information regarding the data on the page, being something external to the object it encodes. The same thing can be said for the use of images; for example, you can see in Figure 9 the appropriated tags used to describe an online image in the early HTML documentation, dated 1993.[38]

As we can see in Figure 9, the image should be a "small image" or "icon"; it is not possible to insert large images. The SRC indicates the URL, ALIGN indicates the visual display, and ALT indicates "alternative text," which is "optional" and is the only place where additional metadata (without a semantically specific tag) can be added. These tags equate to all that a "digital image object" was on the World Wide Web in the year 1993. Then, in 1994, HTML 2.0 was produced, followed by the draft of HTML 3.0 in 1995, followed by the release of HTML 3.2 in 1997. We can see that,

```
        <!-- To avoid problems with text-only UAs as well as
     to make image content understandable and navigable
     to users of non-visual UAs, you need to provide
     a description with ALT, and avoid server-side image maps -->
<!ELEMENT IMG - O EMPTY                        -- Embedded image -->
<!ATTLIST IMG
   %attrs;                                      -- %coreattrs, %i18n, %events --
   src        %URI;       #REQUIRED  -- URI of image to embed --
   alt        %Text;      #REQUIRED  -- short description --
   longdesc   %URI;       #IMPLIED   -- link to long description
                                        (complements alt) --
   name       CDATA       #IMPLIED   -- name of image for scripting --
   height     %Length;    #IMPLIED   -- override height --
   width      %Length;    #IMPLIED   -- override width --
   usemap     %URI;       #IMPLIED   -- use client-side image map --
   ismap      (ismap)     #IMPLIED   -- use server-side image map --
   >
```

Figure 10. Specification of an image in HTML 4.0. http://www.w3.org/TR/1999/REC-html401–19991224/struct/objects.html#edef-IMG.

gradually, further tags are added as the original tags are refined. HTML 3.2 introduced tables, applets, text flow around images, subscripts, and superscripts.[39] We can compare this with the later version, HTML 4.0, as recommended by W3C in 1997 (see Figure 10).

We can see that there were some improvements made in HTML 4.0 (or perhaps one can say it is a better "form"). Many more tags are available, such as those that specify the size of the image. We can see that it is no longer limited to "small images" and "icons." The information is nevertheless still very limited, and it is nearly impossible for the computer to be able to identify what the picture is really about. One can still fill in <alt> to provide a short description of the image, although the computer would not be able to understand this unless it were able to interpret natural language. In fact, throughout the script, the term "object" is taken for granted without any explanation. There are two interesting tags we should pay attention to here: "usemap" and "ismap." These tags equate to two different types of image maps, allowing further specification of what the image really is by linking an intended part of it to another URL. "Ismap" is a server-side image map; it is only designed for very old browsers that do not recognize "usemap" (which is a user-side image map). The image map refers to those of its relations that are external to the image itself, whereby we may begin to notice that the individual does not exist within its own terms but is always related or linked to something else external. Above all, however, the most critical aspect of HTML 4.0 is that it is fully integrated with Cascading Style Sheets (CSS), allowing a more advanced format definition and the presentation of a web page. Objects (both texts and images) can then be described in terms of markups, which make explicit their meanings, and can now be further formatted in terms of their appearances. We should also recognize that this is the process of objectification *as* concretization. The late 1990s saw the increased emergence of multimedia data in the forms of Shockwave, Flash, MP3, and so on, which naturally demanded an improved means of representation. Without these descriptions, the search engines would not be able to locate the data, and the data would eventually dwell in the dark corners of cyberspace, to remain forever lost and unknown. This outlined problem (the lack of semantic meaning) would later be addressed by the recommendation of eXtensible Markup Language (XML).

XML and the Rise of Web Ontologies

XML was also an adaptation from GML, or rather SGML with a simplified syntax. The development of XML was primarily established to improve the lack of flexibility of HTML and to lower the barrier of SGML, which was found to be too heavy to be used on the Web. XML also plays a significant role in what I mentioned earlier as the "time of relaxation." Around the year 2000, there was the dichotomy between the Microsoft Windows (.Net) and Sun Java (J2EE) frameworks; XML subsequently formed frameworks external to these, providing a bridge between the two technologies.[40] In comparison with SGML, on one hand, XML placed some stricter rules on syntax, for example, denoting an unclosed tag as a mistake; on the other hand, it discarded some of the complicated syntaxes of SGML. One example of these differences is that for SGML, a DTD must be "valid," whereas for XML, any well-formed data with a proper tag syntax will be allowed (even without a DTD). This makes XML easy to use, even for those who are not already familiar with the SGML specifications. A user would easily be able to create an XML file describing an image according to common sense and previous knowledge. See Figure 11.

If we compare this with the earlier example of HTML 4.0 (Figure 9), XML can achieve a lot by restricting the user-programmer to providing information on the objects according to what is in demand or what is

```
<image>
        <title></title>
        <author></author>
        <link></link>
        <camera></camera>
        <location></location>
</image>
```

Figure 11. Simple example of an image in XML format.

considered to be useful. In the case in which a computer program is written and designed to analyze data, it is subsequently able to track down information such as who retrieved this photo or image and where it was taken. Such information can be very useful for information retrieval, enabling the programmer to extend the XML by adding more attributes for a more detailed description in simple terms. Thus the description might state who is represented in the picture, when it was taken, and so on. This is the fundamental idea of XML, though there are many other technical details that will not be covered here. In terms of objectification, XML goes much further than HTML by imposing a *more flexible yet stronger form.* Parallel to this, it is able to share the restricted semantics with any ordinary user. In 2000, W3C recommended XHTML (which is a combination of HTML 4.0 and XML 1.0) to adopt the HTML set of attributes toward structural and visual representation and include the syntax of XML for structured content presentation. For example, "namespace" (which can be understood as a prefix) is added, so prefix1:cat and prefix2:cat can be distinguished despite the fact that they share the common suffix "cat." With these tags, a computer program will be able to extract these data from the web page automatically.[41] What is interesting here, and also relevant to our discussion of hylomorphism, is the evident failure of XHTML2, which was introduced in 2002 and officially "died" in 2009. XHTML2 has been described as "a beautiful specification of philosophical purity that had absolutely no resemblance to the real world";[42] however, its fundamental problem was that it was simply too distant from the technical reality. It was neither backward compatible nor compatible with the common practices of developers. Because only a few developers used XHTML2, its death and disappearance did not cause much affect.

In April 2011, W3C introduced HTML 5.0, a single language that integrates both earlier versions of HTML with XHTML. They introduced two very significant changes that are relevant to our discussion here. The first change was the introduction of Application Programming Interfaces (APIs, for example, audio and video players, drag-and-drop APIs), which extend HTML from a textual representation to some forms of pseudo-software. The second improvement made by HTML 5.0 was to introduce a further series of <object> diversities, including <audio>, <video>, <canvas>, and so on. Many more attributes were added to enable better grasp of the objects, or rather, we may say, to achieve a greater "objectification" of data. Let us consider the example of <img> in HTML 5.0. One is now able to indicate the appearance of an image according to its status as either "unavailable," "partially available," "completely available," or "broken," as well as to display the downloading status when showing the images.[43]

We have noticed that within digital objects, the *concept of form* continues to serve as a technical tendency within computing, although it is now *standards* that have become universal. Forms are abstract schemes, and standards are concrete objects. We must also bear in mind the other aspects of standardization that are political and economic. First, it is an enforced technical process that pursues the compatibility of computation on global scales, and second, it is also a marketing strategy that builds up networks of partners and alliances. We focus only on the first aspect here. Because XML is freely extensible, some programmer may use scheme A to describe an object, whereas another may prefer scheme B, the result being that there will be a lack of objectivity. Objectivity in this context should be understood to refer to the character of elements that come from an object itself and remain universal to the observers. In science, for example, an objective method and an objective mode of observation exclude all forms of subjective and psychological interpretation. This understanding of objectivity bears within it a paradoxical relation to universality. We have already discussed the first meaning of universality, in the context of the separation of content from form. Being universal, the form becomes a shared framework for every machine, whereby its modification may lead to incompatibility. So to disclose a form without variation, it must be seen objectively. This highlights one of the problems associated with the freely extensible XML. As XML guarantees the format and validity of the form, it does not guarantee the objectivity of the scheme (the set of tags used, in this case). This objective–universal correlation can be contrasted against another kind of universality, one that allows for differences. Berners-Lee was certainly not unaware of this contradiction, as he compared this second understanding of the universal with the Unitarian Universalism religion.[44] Unitarian Universalism incorporates doctrines from all religions, creating a space for differentiation. For Berners-Lee, this is one of his key design principles for the Web, for example, in his proposal concerning lightweight HTML and low-level XML. The minimization of forms allows for further extension and adaptation.

This ambiguity becomes obvious when XML is conceptually modified into an ontology. In an article published in *Scientific American* in 2001, Tim Berners-Lee and his collaborators proposed the idea of the semantic web as a place where, they envisaged, all objects are represented by standard ontologies. These ontologies, based on XML syntax, regulate the semantic meaning of the objects in a way that enables machines to understand and manipulate data. Each object–predicate is identified by a unique URL, which serves as an ID within the digital milieu. So not only do the objects have identities, but their components or predicates also have identities and are thus subject to control and manipulation. Berners-Lee and colleagues began with an imaginary scenario: that Pete and Lucy's mother needed to see a specialist on a regular basis. Their semantic web agent (a computer program that is capable of analyzing ontologies) can tell them the location of the hospital, the best way of getting there, how to make an appointment with the clinic's agent, and how to reschedule their own work to fit in with their mother's appointments. Berners-Lee continues to describe the semantic web as follows:

> The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. Such an agent coming to the clinic's Web page will know not just that the page has keywords such as "treatment, medicine, physical, therapy" (as might be encoded today) but also that Dr. Hartman works at this clinic on Mondays, Wednesdays and Fridays and that the script takes a date range in yyyy-mm-dd format and returns appointment times.[45]

What exactly is the difference between ontologies and XML? A technical explanation expresses the following: (1) "an ontology differs from an XML schema (which describes the structure of a XML document) in that it is a knowledge representation, not a message format" and (2) "one advantage of OWL [Ontology Web Language] ontologies will be the availability of tools that can reason about them."[46] These two points of comparison require further discussion. Knowledge representation here doesn't mean mere representation but is necessarily objective, so that what it presents can be recognized as an object instead of a set of textual messages. To reconcile objectivity and the two differing forms of universality, two presuppositions are to be made: (1) that there is an objective representation of things and (2) that their translatability can take place in between two representations of things, allowing the object from context A to be translated into an object from context B. This translation process is simply the translating of vocabularies and prefixes. If we stop to consider this for a few seconds, we realize that translation would be impossible without the second presupposition. What dominates here is the concept of objectivity as universal. Facts can only be meaningful when they can be subsumed to forms, whereby they can be regulated and calculated. Let us now examine the example in Figure 12 of an image in an ontology-driven information system. The figure shows a sample of data that were extracted from Flickr in 2007 (this is just a small sample of the metadata contained by this chosen image;[47] these data were extracted using Flickr's public API function [Flickr.photos.getInfo]).[48]

The extracted data sample appears to be relatively large (considering it was already obtained a few years ago, it can be larger today); "what an image is" is apparently much greater than the sum of the definitions and descriptions by which HTML 4.0 designates an image. We easily see that the information given here is much more extensive than that which we derive from actually looking at a picture and includes geodata, camera information, the time of uploading, different reference IDs, friends' information, and so on. We can even see that the image object simultaneously embeds various camera objects, author objects, location objects, and so on. An object is therefore determined not just by a single form but by multiple forms (or by its ground, to echo Simondon). We shall return to the concept of ontology and relationality in the subsequent chapters of this book; our focus for now is simply to grasp the process of individualization—which is not simply the concretization of the object but also the creation of technical associated milieux without which it cannot function. Throughout the concretization process from GML to web ontologies, a digital object can be described in a more and more detailed manner, at the same time

comments:	1
dates:	
dateuploaded:	8/19/07; 2:44:43 AM
lastupdate:	8/19/07; 2:44:43 AM
posted:	8/19/07; 2:44:43 AM
taken:	8/18/07; 10:44:43 PM
takengranularity:	0
description:	Sent from my iPhone
editability:	
canaddmeta:	0
cancomment:	0
farm:	2
geoperms:	
iscontact:	0
isfamily:	0
isfriend:	0
ispublic:	1
id:	1166257196
isfavorite:	0
license:	5
location:	
accuracy:	15
country:	United States
county:	Santa Clara
latitude:	37.444293
locality:	Palo Alto
longitude:	-122.160591
region:	California
notes:	
72157601607070993:	
author:	22221172@N00
authorname:	scriptingnews
h:	20
id:	72157601607070993
title:	Blue Chalk Cafe

Figure 12. A sample of data extracted from an image on Flickr.com.

establishing material connections over a broader milieu across further platforms and interfaces. The ontologies are then continuously formatted through the Resource Definition Framework (RDF) (proposed by the W3C). RDF is also based on the syntax of XML, thus having a logical form. An RDF statement follows the rules of first-order logic, such as in the following coding:

<subject>+<predicate>+<object>

This simplicity allows for an inference language and a succession of logical operations on a machine level. The transition from XML to a more logi-

```
w:                68
x:                280
y:                14
originalformat:         jpg
originalsecret:
owner:
location:               USA
nsid:             22221172@N00
realname:         Dave Winer
username:         scriptingnews
rotation:               0
secret:
server:                 1007
tags:
barcampblock:
author:                 22221172@N00
id:               380915-1166257196-13743477
machine_tag:      0
raw:              barcampblock
heatherharde:
author:                 22221172@N00
id:               380915-1166257196-2504570
machine_tag:      0
raw:              Heather Harde
techcrunch:
author:                 22221172@N00
id:               380915-1166257196-3057
machine_tag:      0
raw:              TechCrunch
title:            Heather Harde, TechCrunch CEO
urls:
photopage:   http://www.flickr.com/photos/scriptingnews/1166257196/
visibility:
isfamily:               0
isfriend:               0
ispublic:               1
```


cally defined RDF is a significant move toward an AI-motivated Web. In 2002, another standard OWL was introduced to improve the performance of logical operations. OWL is precisely the language developed by the W3C for ontology construction. There are three versions of OWL, each differentiated according to its different purposes and complexities per use. The highest and most sophisticated level of OWL is a logical language that formulates variables such as class, property, relation, and cardinality. The use of OWL will benefit from the "availability of tools that can reason about them," or in the words of Berners-Lee, the machine can "pretend to think."[49] The relations between OWL, RDF, First Order Logic (FOL), and Description Logic (DL) are further addressed in chapter 5.

To summarize the preceding discussion of the individualization of digital objects, we have recognized that this process embraces three key concepts: *universality, interoperability,* and *extensibility.* These are all, coincidentally, synonyms for "objectivity." Yet we can see that this objectivity is in fact in the constant process of evolution or individualization. This objectivity is not limited to human understanding but also requires machine interpretation. The discussion around the objectification and individualization of the "digital milieu" has only very recently entered a more mature phase. *Horizontally,* we can see that forms have developed from GML (to allow compatibility between programs within a machine) to ontologies (across the Internet, in between machines), a process that gradually involves a greater number of objects, machines, and users to maintain its functionality and stability. We can also approach the associated milieu as a measurement of interoperability and compatibility here. *Vertically,* we can see that digital objects are always within a process by which they gradually become more concrete and individualized. HTML is simply a formatted text file full of data, whereas RDF is a complicated document coded with advanced programming and logical developmental capacity. The RDF- or OWL-formatted ontologies thereby become similar to an object in object-oriented programming (OOP). OOP has three important properties: abstraction, encapsulation, and inheritance, whereby a class can be overridden to generate new classes, which subsequently inherit certain properties and functions from the parent class. We can identify all these properties within the current concept of web ontologies.

The genesis of digital objects forms the beginning of an investigation into the dynamics of these objects, aimed at developing the scope for a better understanding of the meaning of this new genre of industrial objects. Following on from Simondon, we can apply the concept of genesis to digital objects, while additionally discovering new dynamics that we previously would have ignored and dismissed as mere objects. The genesis of digital objects is the process of concretization and materialization, first of forms, second of explicit relations and connections between objects. We can also see this as an evolutionary process of interobjectivities in contrast to intersubjectivities, which we further elaborate in chapter 4. Now at the end of this chapter, we have arrived at the creation of ontologies after a discussion of forms as a general technical tendency. Now we shall ask the question, where do these ontologies come from? and seek to understand what is involved in the word and concept of *ontology* detaching from its metaphysical context and becoming purely practical. In the next chapter, these questions are addressed in greater depth through an investigation of the theories of Brian Cantwell Smith, Edmund Husserl, and Martin Heidegger concerning objects and ontologies.

This page intentionally left blank