



A Proposed Framework for Cellular Evolution

Albert D. G. de Roos

DOI: 10.33014/issn.2640-5652.1.1.deroos.1

Abstract

This article shows how a good understanding of the design principles that underlie life is crucial in understanding its evolution. It gives concrete examples on how designs can be applied to evolution in the same way an engineer uses designs in creating building or bridges, or a software engineer that designs programs that can evolve. If we apply the notion of a design for evolution and realize that it is in its basis an evolving system of molecular machines, we can start to reverse engineer evolution and understand life. This article shows scenario's for the origin of life and cellular life and how eukaryotic cells evolved to multicellular organisms. The driving force for evolution would be the intrinsic capability of the molecular machines to evolve and this basic tendency to evolve its ultimate goal. The scenarios that can be deduced following a design framework are drastically different from the theories that have been proposed so far. Based on the premise that that evolution can be modeled on a design framework, we can deduce that life evolved inside-out and that ontology reflects phylogeny on a molecular basis

1 Evolving Machines

1.1 Evolution of the eukaryotes

In order to understand life and its evolution, we first have to see how living organisms function. The cell is the main building block of all organisms and is in itself a small molecular machine, illustrated nicely by the cell cycle. In higher multicellular organisms, the cells have specialized and formed into organs and limbs but the cell remains the main building block. The mechanistic organization of multicellularity can be easily seen when we look at development from a fertilized egg to an organism. The evolution of life starts with the evolution of the cell, specifically the eukaryotic cells that contain a nucleus and forms the basis of all higher life. This paper concentrates on the evolution of the eukaryotic cell, which includes not only all plants and animals, but also protists such as algae and yeast.

1.2 Cells are molecular machines

The basic premise underlying the study of design in evolution is that the eukaryotic cell is basically a small molecular machine where the components are formed by molecules. Cell membranes are formed by lipid bilayers, the cytoskeleton is made up of protein fibers, our genetic material is made up of a double stranded helix. There is machinery that can assemble these materials illustrated by cell division showing a carefully orchestrated sequence of mechanical events. In this process of cell division, DNA has to be replicated, the nuclear membrane dissolved, chromosomes condensed and aligned, the replicated chromosomes need to be pulled apart, the nuclear membranes reformed and the physical split of the two daughter nuclei. It is not hard to see this as a purely mechanistic process with many checkpoints, regulators, machinery, feed-back and feed-forward loops and self-assembly of machinery. The question for evolution is how this system evolved and how the molecular components were assembled over time.

1.3 Development represents a mechanistic cycle

Just as the cycle of individual cells, the development of an organism from fertilized egg to the adult organism is also a purely mechanical process. Based on the genetic blueprint containing the information for the protein components and the assembly of the organism, development consists of self-assembly, feedback and feedforward loops and self-learning systems. The sequence is put in motion by the fertilization of the egg until the adult organism where, several genes have to be activated at specific point in time of development. In humans, development involves the coordinated and timely activation of 100,000 genes over a period of more than 50 years. This process is tightly regulated with checkpoints and many other regulatory processes that check certain condition before progressing cell division and organ formation. The evolution of such a multicellular system adds another dimension to evolution, because we look at the assembly of the organism over time.

1.4 The life cycle as vehicle for evolution

The life cycle of an organism is defined from the single cell or fertilized egg, all the way through development and ultimately to the rise of another new organism by fertilization. Every life cycle has to end with the birth of a new organism and the organism should always reach the reproduction phase in order to complete the cycle. Each step or life cycle has the potential to add extra functionality and thus evolve, but throughout evolution the development of fertilized egg to mature organism has to be fully functional. Any functionality that would cause a cell or an organism to cease functioning and to interrupt the reproductive cycle would stop to exist. This essential functional continuity is an important (design) constraint for evolution implying that all the steps in evolution must each give rise to a fully functional organism or cell system. Drastic redesigns are impossible in evolution as they would certainly break the functional continuity principle.

1.5 Incremental development

In line with the principle of functional continuity, evolution can be seen as the incremental expansion of an existing system. The components of the system are re-assembled during development and with every life cycle, there is the potential for new functionality. Mutations in the genetic material can give rise to new sorts of protein that can give rise to new components of the system. All the changes that have occurred in evolution and that give rise to the specific developmental pattern that is encrypted in the genome of the cell, for eukaryotic cells mainly in the chromosomes in the nucleus. In this sense, evolution can be regarded as a molecular machine (an organism) that can self-assemble itself in each life cycle with the potential to add new functionality on top of the existing functionality. Since the blueprint of the self-assembly is contained in the genome (DNA), the expansion of the blueprint in time is what we call evolution.

1.6 The software development life cycle

We can compare the stepwise evolution of the molecular machine we call life with the step-wise development of a software program. With each cycle in software development some extra programming for new functionality is done resulting into a new functional release. We want to create of system with many functional modules that has evolved from previous less complex stages. If we model evolution on building software, we would start with some simple code that creates a program but that we build incrementally.

With each software cycle we would create a working product unto which to build further and each iteration of the software should be fully functional. In software development, we work with concrete design and design patterns in order to build complex software systems that continue to keep working with increasing complexity. In order to use the analogy of software design to evolution, we have to specify a similar framework or model that will represent evolution.

1.7 A concrete design paradigm for Evolution

The basic premise in this article is that a design framework similar to the concepts used in engineering and exemplified in complex IT-systems and software development can be used to model evolution. The working hypothesis in the research presented here is that evolution can be modeled on the software design methodology 'design-by-contract', the basic design underlying complex software systems. A strict implementation of design-by-contract is enforced by the condition that earlier 'contracts' must always be respected otherwise the premise of functional continuity will be breached. Evolution of such a system can then take place in different manners for instance by the creation of completely new functional modules build on top of others, or by hierarchical extension of existing interfaces. Based on the same requirements, basically functional continuity in evolving systems, the concrete design patterns used in software development can therefore also be used for modeling evolution.

2 Software Design Patterns

There are many similarities between software design and evolution. Except for a more general architecture of incremental development as we have seen as the basic principle behind evolution, several design patterns that are relevant for evolution can also be discerned in software development. Here, some of them are described. Mostly, they represent common patterns in engineering that are formally conceptualized in software design.

2.1 Modularity

In many ways, the requirements for the evolution of life are similar to those of modern software design. Software systems often start small but soon extra functions are needed

or existing ones need to be changed. As the system grows by adding new functions, it becomes more and more difficult to change the system. Changes in one part may have an effect on other parts and can therefore have unintended consequences. This makes that we want to have a system that is robust yet flexible. In computer science, these requirements are usually implemented by having modular design in which functional modules communicate with each other through interfaces. By hiding the complexity within each module, we can reduce the overall complexity of the system. Modularity is something we also expect in evolution as it allows independent evolution of subparts.

2.2 System expansion through design-by-contract

The problems of increasing complexity in a system that needs to be robust, flexible and resilient have been addressed in software development by using the design-by-contract methodology. It works by dividing the system in functional modules that communicate with each other in a specified way, a sort of contract that both parties need to adhere to. The only thing a module has to know about other parts of the system is through this interface and as long as you do not change the way you communicate, each function can evolve independently. The reason that established interfaces cannot be changed is because downstream processes rely on that interface. If you change them, all processes that are dependent on it, may not work anymore. For evolution that we also see as an expanding system of functionalities, it means that all processes or functional modules that are added later in evolution will be dependent on earlier interface and therefore difficult to change.

2.3 Encapsulation

In software design, the term encapsulation is used when a functional layer is added as a layer on top of the old layer without replacing the old layer. This is for instance very useful when dealing with legacy systems that function correctly and that we do not want to replace. For instance, instead of using the old command-line interface in MS-DOS, we can introduce a graphical Windows layer on top that will translate the commands for us. Another example is the internet itself that uses in its basis (and in its origin) a simple protocol that sends binary data over a network. Once that was established, other layers were added to route the data through more complex networks and to give instructions to higher level systems. Thus, by layering functionalities on top of each other, we can create complex systems that are

robust as long as we keep existing (lower level) interfaces intact. For evolution, encapsulation would provide a way to keep existing functions by wrapping them with new, more modern functionality.

Legacy systems System that are developed using the design-by-contract methodology leave older interfaces intact causing the system not only to be robust but also inflexible. Older parts of the system cannot be changed anymore because of the dependencies of new functionality upon these older parts of the systems. Many software systems contain old subsystems that were developed some long ago but could never be changed because the code is essential and changing it could cause them not too work anymore because of the many dependencies. In other words, the functional continuity of the entire system could not be guaranteed by redesigning the core part. An example is the old Cobol code for banking systems that were developed in the 70s but are still in use until a complete redesign is warranted. In evolution, there is no redesign possible as it would breach the functional continuity that is needed. The risk of changing them for the stability of the systems prevents changing them.

3 The Origin of Life

3.1 The central dogma of Biology

In analogy with software development, if we want to unravel the origin of life we should first define the functional modules and their interfaces, in this case the molecular counterparts of modules and interfaces. As the basic structural entities underlying life, we can discern the double-stranded DNA as the carrier of the genetic information, the single-stranded RNA that can both function as a ribozyme and as a template for translation into proteins. The molecular machinery for DNA replication, for transcription and for translation can be considered the functional modules. The interfaces between these three functional modules are the double-stranded DNA helix with the four bases (ATCG), and the single-stranded RNA with the four RNA bases (AUCG) as template for translation into proteins. This configuration is called 'The central dogma of Biology'.

3.2 Evolution of the dogma of Biology

If we look at the dependency and the relations between the functional modules of the central dogma, we can deduce the sequence of evolutionary events. DNA is transcribed into

RNA that functions as the code (interface) for the proteins translation, so double-stranded DNA had to be first to have evolved. From this initial DNA, single-stranded RNA was derived which first functioned as catalytic RNA. Using the catalytic characteristics of single-stranded RNA the first proteins were made using single-stranded RNA (mRNA). Thus, we can directly reverse engineer the steps in evolution from the developmental steps, just by studying the interfaces and their dependencies. The general rule here is that we look at the flow in development steps (dsDNA to ssRNA to protein) to deduce the evolutionary steps.

3.3 Double-stranded DNA

All life is based on double-stranded DNA and it is therefore important to see how this double-stranded DNA could have evolved in the first place. DNA is composed of nucleotides that have the tendency to stick to other types of nucleotides. Nucleotides can also form covalent bonds between them, so that they can form strands. The combination between single-strand formation and hydrogen-bonding between the nucleotides can make that the strands bind to complementary strands of DNA forming a double-stranded DNA. Thus, the double-stranded DNA can be formed based on the basic characteristic of nucleotides to form covalent bonds between nucleotides and hydrogen bonds between complementary nucleotides. Nucleotides and their capability to form covalent bonds and their intrinsic tendency to form complementary strands can form the first interface for life. Although it is thought that RNA would precede DNA as a prebiotic molecule, they have similar characteristics and a later change to DNA would not affect any interfaces based on RNA.

3.4 Replication of DNA

Replication of DNA is based on the characteristic of DNA that it can form its own template, but in order to replicate the strands have to be separated. The covalent bonds between the nucleotides that form the single-stranded DNA are quite strong, but the hydrogen bonds between different strands can be broken more easily for instance by heating the double-stranded DNA. This process is also called melting of DNA and yields two separate strands of DNA. The melting of DNA can be done by an increase in temperature, but also through the action of proteins. As proteins were not present at the origin of life, melting by an increase in temperature is likely to be the first way of separating the two strands for replication. The day/night cycle may have provided the necessary heating and cooling of an original

DNA mixture for early life, similar to the PCR reaction in which DNA is replicated in a laboratory. Even though the replication process is now facilitated by proteins, its basic mechanism has not changed.

3.5 Transcription of DNA into RNA

If we look at the current events in the life cycle of an organism, we see that apart from serving as its own template for replication, single-stranded RNA is also generated from the same template. Mechanistically in an abiotic cycle, the formation of single stranded RNA can be relatively easy accomplished by partially melting of the DNA and the 'replication' of the exposed single-stranded DNA. This process is not fundamentally different from what happens in transcription where the DNA is partial single-stranded (in the transcription bubble) so that genes can be transcribed. Single-stranded RNA is directly transcribed from the double-stranded DNA and can function by itself as a catalyst (hence the term catalytic RNA) or as a template for proteins in a separate process. Throughout evolution, the transcription process has not been changed fundamentally and the interface (double-stranded DNA that codes for single-stranded RNA has not changed.

3.6 Protein translation from mRNA

The input for protein translation by the ribosome is single-stranded RNA, also called messenger RNA. The ribosome is the machinery that does the translation and adds specific amino acids based on the genetic code to a growing chain of protein. As we have seen, ribozymes can also be derived from double-stranded DNA and ribozymes could thus have been involved in the first translation to proteins. In fact, the ribosome has many RNA components and can be considered a ribozyme in its core. For evolution, this also shows a logical sequence of events, where first ribozymes were derived from the double-stranded DNA template which formed the machinery to start using single-stranded RNA as a template for proteins. Here we see that the same template (ssRNA) can be used to evolve different functionalities while keeping existing interfaces intact. The evolution to protein generation can be seen as an independent functional module.

3.7 Evolution of the dogma of Biology

If we look at the dependency and the relations between the functional modules of the central dogma, we can deduce the

sequence of evolutionary events. DNA is transcribed into RNA that functions as the code (interface) for the proteins translation, so double-stranded DNA had to be first to have evolved. From this initial DNA, single-stranded RNA was derived which first functioned as catalytic RNA. Using the catalytic characteristics of single-stranded RNA the first proteins were made using single-stranded RNA (mRNA). Thus, we can directly reverse engineer the steps in evolution from the developmental steps, just by studying the interfaces and their dependencies. The general rule here is that we look at the flow in development steps (dsDNA to ssRNA to protein) to deduce the evolutionary steps.

3.8 The first legacy system

We see that although the implementation of the functional modules has changed, their interfaces have not changed. DNA replication is now much more efficient than in the early years with many more proteins and cofactors involved, but the basic interface (template-based replication) has not changed. Although there were more modern and efficient functions implemented over time, its basic interfaces were kept intact. Once the central dogma was established all subsequent functions, whether it is bacterial growth or the formation of multicellular organism would depend on the formation of catalytic RNA and proteins. In all life forms, the central dogma still holds and can thus be considered the first legacy system. These basic interfaces simply could never change because it would break the rule of functional continuity. Everything else is based on these original and additional layers of functionality have been built on top of this, as you would expect in a system that is based on a modular design based on design-by-contract.

4 Early Genome Building

Design-by-contract divides a system into functional modules and their interfaces. The modules perform a certain task in the system and communication between modules is through defined interface. Design-by-contract states that these basic interfaces cannot be changed once established and we applied this concept to the evolution of the central dogma. Here we see how a complex genome can be formed based on the central dogma while applying design-by-contract and the rule of functional continuity.

4.1 Exon concatenation

If we look at DNA and its structure, there is a clear modularity at the gene level. The gene as a functional module that codes for a protein and its interface consists of the promoter with the start codon and the introns and the exons. Within the gene, the different exons can also be regarded as functional modules as they can be combined to give different proteins when spliced into an intronless mRNA. The splicing machinery as functional module has a clear interface: single-stranded RNA with introns and exons are used as the input and messenger RNA without introns is the result. Intron splicing, or exon concatenation is a way to form complex (multi-exon) genes without changing the original interface. Single-stranded RNA is still derived from the double-stranded DNA and the ribosome still works with the same template. The concatenation of exons creates new multi-exon proteins while leaving all other systems intact ensuring functional continuity.

4.2 Exon and gene shuffling

Once the system has evolved multi-exon genes, further combinations of a limited number of functional and structural exon modules can create a large set of different proteins. Introns and other non-coding sequences provide recombination points that make the shuffling of genes and exons feasible. Higher level shuffling can be performed when complete protein modules composed of multiple exons are shuffled made possible by the modular structure of genes and exons. The exons are recognizable in the strands of DNA by a nucleotide sequence in the intron that demarcates the exons and the entire exon modules would function in another gene. Transfer of parts of genes with their promoter regions that demarcate the start of a gene, allow transferred genes to be transcribed without needing extra signals. The modular structure of genes thus allow for quick expansion of the genome and its concomitant expression of proteins.

4.3 Posttranslational modifications

The ribosome translates messenger RNA into proteins and is unaware whether the mRNA contains introns. Alternative splicing and exon shuffling were able to diversify the number of proteins without affecting protein translation itself. Also after translation, the cell has evolved mechanisms to expand the total number of functions of proteins by post-translational modifications, for instance by adding several chemical groups. Using the original machinery that transcribes mRNA from DNA and that translates mRNA to

proteins, an enormous diversity of proteins can be created without changing these basic steps. This is done by leaving the earlier interfaces intact in line with the design-by-contract paradigm and adding extra functionality on top of the existing system.

4.4 An expanding molecular system

In eukaryotic cells, there are mechanisms for splicing and alternative splicing, but also for gene recombination and gene hopping. These mechanisms can be seen as the vehicles for the evolution of new proteins. Active mechanisms for gene recombination could have quickly generated a diverse gene population with only a limited set of functional exon modules. There could also be many other active processes that are involved in an active expansion of the genome during evolution. Crossing-over during meiosis in the germ line cells is mechanism that creates recombinations. There are enzymes (telomerases) that are involved in extension of the chromosomes and they could have created space for new proteins. Facilitated by the modular structure of the genome, active recombination mechanisms can drive genome while maintaining functional continuity. In other words, due to active gene recombination mechanisms, the genome can be considered to be self-evolving.

5 The Origin of the Nucleus

The nucleus is a membrane-enclosed organelle found in all eukaryotic cells. It contains most of the cell's genetic material, organized as multiple long linear DNA molecules in a complex with a large variety of proteins, such as histones, to form chromosomes. The genes within these chromosomes are the cell's nuclear genome. The main structures making up the nucleus are the nucleolus, the nuclear envelope and the nucleoskeleton (which includes nuclear lamina). The structure and relationship between these functional units and the nucleus can give insight in the stepwise evolution of the nucleus.

5.1 The nucleolus as first replicating unit

In the evolution of the nucleus, we start with replicating strands of DNA that can produce ribozymes and proteins as we have discussed in the previous chapter. We can relatively easy start extending this system into a more complicated structure by adding a microenvironment of protein fibers, DNA, ribozymes and ribosome components. Pro-

teins can give the DNA structural support and that create a microenvironment that would facilitate duplication. The nuclear matrix, a scaffold of proteins could serve as such a protective layer around the DNA and would lead to a structure that we now call the nucleolus. Such a system that is based on the protein components that the early translation machinery can produce would be the first step to a primitive cell. This nucleolus would be composed of a pool of DNA that produces its own individual environment.

5.2 The appearance of the nuclear lamina

From the nucleolus as the first compartmentalized structure that contains DNA, the next layer would be the extension of nuclear matrix proteins around the nucleolus. The nuclear matrix is a scaffold of protein fibers that surrounds the nucleolus and this would create an additional (protective) protein layer that could create the environment for additional processes. On top of this fiber scaffold, the nuclear lamina can be formed as a distinct protein scaffold that can form a semi-permeable barrier resembling the current nucleus but without the nuclear membrane. At that moment, a primitive cell is formed that consists of a replicating chromosome embedded in permeable, multiple layers of protein scaffolding. Since the proteins for the nuclear lamina originate from DNA genes, the early genome can develop its own environment.

5.3 Self-assembly of lipids on protein scaffold

Lipids play an important role in the biology of the cell and many chemical reactions take place with the help of lipids. Also, lipids have the ability to form vesicles and membranes to form a specialized microenvironment. The evolution of lipid generating proteins could lead to microvesicles that could facilitate replication and these vesicles could be associated with the nucleolus and the nuclear matrix. At a high enough density and by the expression of certain nuclear lamina proteins, these vesicles could fuse on top of the nuclear lamina and form the first nuclear membrane. The evolutionary steps in the formation of the nucleus can be seen at cell division, where the nuclear membrane is dissolved into vesicles and reforms after the cell division. Thus, by expressing a set of proteins, a self-assembly of lipids on the nuclear lamina the nucleus could have formed as the first membrane-surrounded cell.

5.4 The nucleus as an independent organism

The first 'cell' with a membrane in evolution would in our model consist of the nucleus and would precede the eukaryotic cell with a plasma membrane. It would be an independent unit that contained all the information to replicate itself. If the DNA is replicated, this nucleus could physically divide into two nuclei as we see with each mitosis. In this scenario we see the nucleus as an independent unit that can replicate itself into other nuclei since each genome can generate its own proteins for the nuclear matrix and formation of a nuclear membrane. The independence of the nucleus as an independently replicating unit is seen in the development of *Drosophila*. Here, the nucleus of the fertilized egg divides numerous times within the cytoplasm, generating a large number of nuclei within a single cytoplasm. Later, these nuclei organize themselves, mainly guided by an extranuclear network into mononuclear cells. The nucleus can then be seen as replicating itself into two within the cytoplasm and in principle independent from the division of the eukaryotic cell itself.

5.5 The nucleus as a legacy system

The scenario presented here indicates that the nucleus evolved inside-out and started with the nucleolus upon which literary new layers of functions were added that resulting in the nucleus as an independent unit. We then see various structural and functional components in the nucleus that are functionally and structurally dependent on each other. Next to the distinct entity of the nucleolus, we discern the larger nuclear matrix and finally the nuclear lamina as a protein sheath that encapsulates the chromosomes and ribosomes. During the evolution to the eukaryotic cell, the nucleus and all previous processes layers remain intact. The entire nucleus of eukaryotes can therefore be considered a legacy system that functionally stays intact during the entire path of eukaryote evolution.

6 The Eukaryotic Cell

We saw that we can stepwise build up a simple proto-cell that evolved into a self-contained replicating unit: the nucleus as the first real cellular life. The next major step in the evolution of life was the evolution of the plasma membrane that surrounds the nucleus. From the distinct functional components we can deduce the evolutionary events to generate the main components of the eukaryotic cell.

6.1 Formation of endoplasmic reticulum (ER)

The ER membrane is contiguous with the nuclear membrane which means that it shares its membrane and their membranes can be seen as a single continuum. In evolution the transition to the ER could be made when the nuclear membrane started 'blebbing' into membrane lamellae into a special compartment which became later the ER. Membrane blebbing can occur by the expression of membrane proteins and therefore, this step could thus be evoked by the novel expression of a single new protein in evolution. The generation of a separate lipid environment outside of the nucleus would allow many new functions to evolve. For instance, protein translation could be done more efficiently at the ER. The disassembly of ribosome subunits in the nucleus and the subsequent transport of these units to re-assemble at the ER illustrate this process.

6.2 The formation of a cytoskeleton

The eukaryotic cell consists of the nucleus surrounded by the cytoskeleton which provides structure but also a microenvironment for cellular processes. A first step to such an extranuclear environment could be the extension of protein fibers from within the nucleus through the nuclear pores. This would give the nucleus already protection and anchor points for specific energy-rich environments. Later in evolution, when extranuclear (cytoplasmic) translation at the ER could take place, the cytoskeleton could evolve into a more complex protein scaffold. The nucleus would be suspended in its own cytoskeleton that it generated from its genome. Each nucleus would then have its cytoskeletal coat which would define the boundaries of the cell, similar to the *Drosophila* cells in the syncytial blastoderm.

6.3 Formation of the plasma membrane

Starting from a nucleus with an extracellular matrix and an ER, the last step in the evolution of the eukaryotic cell could be the formation of a plasma membrane on top of the cytoskeleton. We see the dependency between the ER and the plasma membrane as the ER and the endomembrane system feed directly into the plasma membrane. Within the eukaryotic cell, vesicles bud off from the ER/Golgi and fuse with the plasma membrane to increase the surface of the cell. A plasma membrane could have formed in evolution in two steps. First vesicles started budding of the ER by expression of new proteins that facilitated budding. Second, vesicles could start fusing onto the cytoskeleton to form

a plasma membrane, also facilitated by the expression of novel proteins.

6.4 Conclusion

The design framework and the resulting new insight in the path of evolution can therefore present a new set of paradigms for new evolutionary theories. The first one is that evolution can be modeled on a design framework which enables us to deduce evolutionary events. The second is that life evolved inside-out and started by the stepwise formation of the nucleus and was followed by the formation of the ER, the cytoskeleton and the plasma membrane.