Systemic A Priori Patterns

Laurent Gasser
Department Tools and Techniques
IRT-SystemX
Palaiseau, France
e-mail: laurent.gasser@irt-systemx.fr

Abstract. **— Relevant patterns in system architecture have a very broad validity. Identifying or leveraging them across application fields is challenging, though. System architects sufficiently knowledgeable in another field have to establish analogies with their own experience to confirm patterns. To let such discoveries happen by serendipity is inefficient. A systemic framework of a priori transformations guides the investigators mining models. From the repeating discoveries emerges a formal pattern. This method is illustrated by the transfer content pattern. Systemically mining for the content transfer pattern brought up variants only seen in field occurrences.**

Keywords-systemic framework; pattern; content transfer; model based systems engineering; MBSE; behavior; SysML; activity diagram.

I. Introduction

Architecture patterns are valuable assets for complex systems design. As inspiring solutions, they improve the products based on them while reducing design effort. Up to now, the production of patterns is often a lucky emergence from random cross-fertilization of application fields. The subject is worth exploring in a more methodical way. We identify a priori patterns in a systemic way, putting a portrait-robot in hands of model miners. Occurrences of these pattern candidates are searched for in various application fields. The observed cases are then consolidated in a field-proven pattern, as illustrated with the content transfer pattern. A systemic catalogue of patterns is brought to systems architects by a method which also helps establishing their field-proven validity.

We expect that creating such a systemic framework will accelerate the discovery of new patterns in the same way that Mendeleev’s periodic table did for chemical elements.

The very definition of a pattern for systems engineering has been debated at first by Cloutier and Verma [1], being close of reuse, heuristics, and framework among other concepts. Since then, it has gained in acceptance [2][3][4]. Results are still far from the ones in computer science. How can it be that systems engineering fails to be more productive in term of defined patterns?

Unlike computer science, systems engineering deals with systems transforming flows of matter or energy in addition to information. The slow emergence of patterns for systems architects may come from the difficulty to reach the expertise level necessary to formulate patterns in several application fields. Good practices are promoted by engineer's education and professional associations. To leverage them across application fields requires multidisciplinary experts, a not-so-common career profile. Often, patterns are strongly influenced by the modeling language or by the application field paradigm. Establishing analogies across these involves ontological translations at several levels. Consequently, patterns are mainly found by serendipity: patterns emerge instead of being hunted for.

We propose to circumvent the difficulty by reverting the quest: instead of asking whether a field-proven pattern could be useful in another application field, occurrences of systemic a priori patterns are looked for in several application fields.

We are aware that practitioners in other fields may be reluctant to adopt patterns found by our method. The effort to qualify and inscribe them in each application field is left to their respective practitioners.

The rest of this paper is organized as follows. Section II shows a general framework of patterns induced by an a priori systemic categorizing of flows and transformations. Section III reports on several field cases of content transfer. Section IV generalizes the content transfer pattern from these cases, with a discussion on its possible variants. The conclusion and acknowledgement close the paper.

II. A Priori Pattern Framework

Being implied in several application fields to detect analog models is a precious experience difficult to organize. Collecting patterns that way is putting one’s faith into serendipity. We aim at exploring potential patterns in a more systemic inventory [5]. We were well influenced by German teaching manuals [6][7] or the Schaum’s series world-wide famous among students [8]. They describe a large number of study cases, often presenting alternative solutions to a single problem, which cover most encountered situations during one’s professional life.

The generalized identification of flow nature and the categories over which they transform is explicitly described by Le Moigne [9] in Figure 5.1 via the transformations that may impact a flow of matter, energy, or information over time, space, or form. Le Moigne agglomerates matter and energy in transformations, though. To us, a transformation may be positive, neutral, or negative in its effect. A positive transformation augments, increases, or intensifies the resulting flow. A negative transformation fragments, scatters, or corrupts the resulting flow. Table 1 proposes a function for each of the cases systemically identified a priori.

TABLE I. A PRIORI PROCESSING TRANSFORMATIONS

Transformation	Matter	Energy	Information
Time	+ collect	+ accumulate	+ learn
	= preserve, conserve	= store	= memorize
	- spoil, decompose	- dissipate	- forget
Space	+ expand,	+ diffuse, propagate	+ broadcast, disseminate
	= transport, anchor	= transfer, distribute	= move
	- confine, condense, concentrate	- concentrate	- isolate, hide, segregate, disturb
Form	+ mold, shape	+ react, refine	+ enrich, augment, decorate
	= replicate	= transform, convert	= translate, transcript
	- erode	- degrade, disturb	- corrupt, sample, simplify, vulgarize

TABLE II. A PRIORI TRANSFORMATIONS OF FLOW NATURE

Transformation	Matter	Energy	Information
Matter	+ purify, combine	+ compress	+ detect, sense
	= transmute, react	= burn, explode	= measure
	- pollute, reduce, lyse	- absorb (shock)	- erase, spoil
Energy	+ centrifuge, distill	+ generate, ignite	+ refresh
	= fuse, melt	= convert, exchange	= detect
	- dry	- decay, degenerate, deteriorate	- perturb
Information	+ cook	+ optimize	+ decode, uncompress
	= actuate	= command, control, pilot, solicit	= process, synthesize
	- screen, filter, sort	- disperse, overload	- scramble, disturb

Table 1 reads as follows: the positive (resp. neutral or negative) transformation (in the cell) applies to the flow (column) over a dimension (row). The boilerplate "<function> is a <effect> transformation of <flow> over <dimension>" applied to the upper left item becomes: collect is a positive transformation of matter over time.

A transformation of nature may convert one input flow in an output flow of another nature. The transformation of flow nature may be positive, neutral, or negative in its result.

Table 2 reads as follows: the positive (resp. neutral or negative) transformation (in the cell) of the input flow (in the row) into the output flow (in the column). The boilerplate "<function> is a <effect> transformation of <input flow> into <output flow>" applied to the lower left item becomes: filter is a transformation of information into matter.

The method to exploit these tables starts by selecting one a priori pattern in the tables. It is then looked for in several application fields. Approaching occurrences are collected and compared. A consolidated pattern emerges from the collected cases and is formalized.

III. ILLUSTRATIVE FIELD CASES

Let us select the middle line of Table 1 as the a priori pattern to be searched for in application fields: a neutral transformation of matter, energy, or information over space.

Industrial systems are full of containers whose content has to be transferred. Would the content be matter, energy, or information, it may flow as a stream (uninterrupted over time) or not (packet by packet). Let us shortly sketch one field case for each combination.

Biomedical devices mix blood with reagents to identify pathogens present in the blood sample [10]. During sample preparation, a precise amount of blood is sucked into the pipette and then flushed into the sample tube. Liquid streams from the blood sample into the pipette, from the pipette into the sample tube: a streaming flow of matter.

A cargo ship enters a harbor to embark or unload fret containers [11][12]. The fret containers are stored on the harbor shore, picked up by a crane which transfers them to and from the cargo ship. Fret containers are taken one by one from an open space (the harbor shore) to another (the cargo ship store-room): a non-streaming flow of matter.

In a thermic power plant producing electricity, the primary circuit extracts the heat from the core to produce high temperature and pressure steam [13][14]. The steam expands then into a turbine coupled to a dynamo (electricity generator). The heat streams from the core source into the steam before being converted into mechanical work (steam expansion): a streaming flow of energy.

Rail transport collects freight cars from their origin to bring them to destination. Since each freight car may have its own destination, they are recombined at a classification yard. Each freight car is uncoupled from its origin train, pushed by a rail tractor through a set of points to reach its destination train and locks to it. To minimize the sorting time, the turnouts are remotely commanded, the rail tractor initiate the motion by a push and let it roll to its destination train. Retarding apparatus along the rails absorb excessive speed by offering resistance to freight car progress. The freight car locks to the destination train without harsh bumping. The tractor impulses kinematic energy to the freight car whose momentum is then controlled to ensure locking to its train: a non-streaming flow of energy.

Television operators broadcast video content as part of their broadcast program or on demand [15]. The video is sent from the television operator to all receivers in an uninterrupted stream of images. Depending on the subscriber contract and order, the content is displayed on television. The operator broadcasts the video in a stream to all receivers: a streaming flow of information.

In large organizations, the information technology department manages the software installed on each computer of the company [16]. When a new release of a software is deployed, a central script copies the new software application to each of the targeted computer. The new software is installed remotely onto each supported computer: a non-streaming flow of information.

Whatever the scale and the nature of the flow, the content of a source container is transferred into a destination container by a transferor.

IV. The Content Transfer Pattern

As the examples of the previous section illustrate, transferring content from one container to another is found in many applications varying in size, purpose, and nature of flow. In this section, the content transfer pattern is generalized from the different occurrences that have been cited in Section 3. These have been found by looking after field occurrences of the middle line of Table 1.

The pattern involves at least two containers and the transferor which transfers the content from the source container to the destination container (see Figure 1). All figures are in SysML [17], the modeling language promoted by INCOSE and dedicated to systems architects. The ‘Content Transfer Pattern’ block is not part of the pattern instantiation: it simply packs the components of the pattern together. The ‘Containers’ and ‘Transferor’ blocks are the physical elements of the pattern. In the biomedical device above, the pipette is the transferor from the blood sample (the source container) to the sample tube (its destination).

The transferor gets the content from the source container and transfers it to the destination container. The ‘Transferor’ provides the ‘transfer’ function, which is directly taken from Table 1. In the power plant example, the transformation is about space: heat is extracted from the core to be lead into the turbine. A ‘Container’ provides two auxiliary services: the functions ‘host’ and ‘release’ content.

Note that the transferor somehow holds content while passing it through. But, this is not considered as its primary function which is to transfer content or let content be transferred. The transferor is connected to both the source and the destination, as in Figure 2. The connection is a flow interface of ‘content’ type: information, energy, or matter depending on the flow nature. This artefact is a fret container (matter) in the context of the cargo ship.

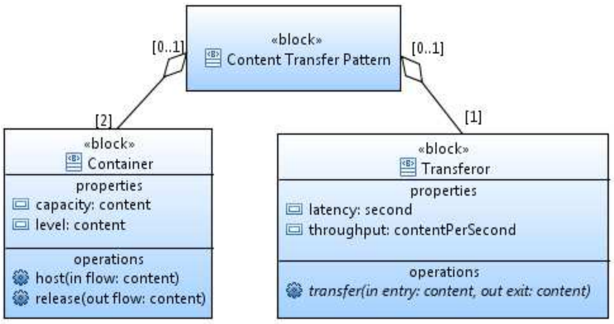

Figure 1. Elements of the Content Transfer Pattern (bdd).

A pattern variant includes a path conducting the content flow fore and aft of the transferor. In another variant, the transferor uses the path to go from one container to the other. The path properties would be its capacity size and length.

By looking at the illustrative field cases, it seems useful to differentiate three specialized transferors according to the propensity of the flow to naturally go from the source to the destination container (see Figure 3). This refinement of the a priori pattern comes from the field cases.

When an adverse potential exists (against the spontaneous transfer from source to destination), the flow from the source has to be pumped up the potential to reach the destination.

When a potential aids to move from the source to the destination, the transferor takes the role of a valve allowing more or less of the flow to pass by.

When no potential difference is noticed, the flow is conveyed from one place to another.

Another point worth mentioning resides in the container behavior. Trivially, a container passively contains its content. Seen as a service provider, the container hosts or releases a content flow. The realized containers always have a limited capacity and a current level of content. A container may be initially full, partially full, or empty. A container is either reversible (a cistern can be loaded and unloaded at will) or not (a non-reloadable battery releases its energy once by irreversible chemical reaction). Three operating modes are defined: empty, full, or containing (see Figure 4).

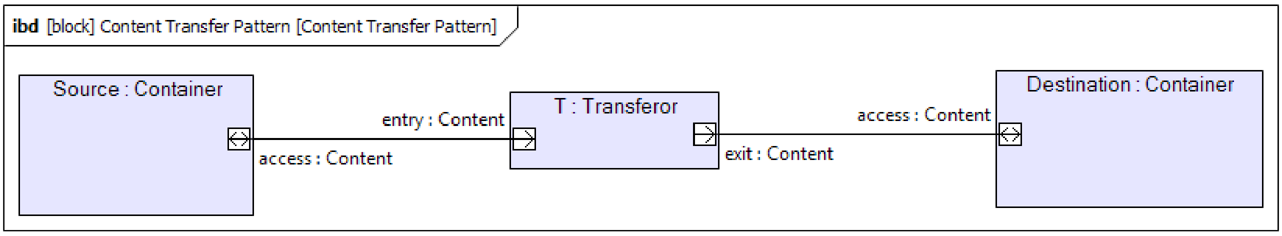

Figure 2. Topology of the Content Transfer Pattern.

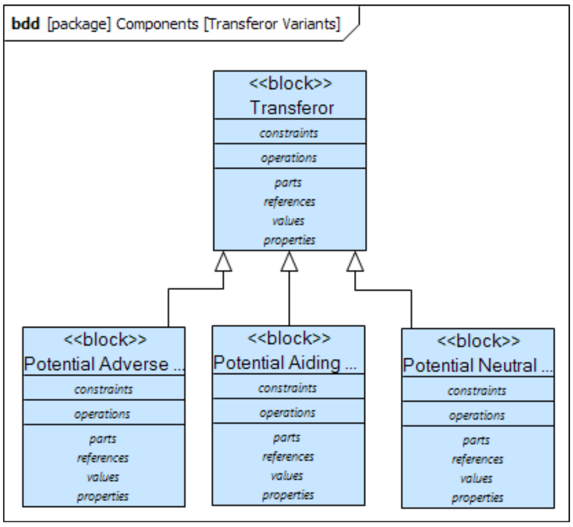

Figure 3. Variants of Transferor

In most cases, the transferor has a unique operating mode, the transfer taking place or not. Reversing the flow in a pump to make it a generator justifies two operating modes of the transferor, though.

The content transfer scenario is rather straightforward (see Figure 5). The two services of the container are used: the ability to accept a content (function "host") and the ability to deliver a part of its content (function "release"). Exceptions occur when the request goes beyond the current content level of the source or exceeds the destination capacity. The transfer content pattern is described in Table 3 using the template proposed in [1].

As is stands, the content transfer pattern can be applied to a wide variety of problems. Electricity distribution has long been a stable situation of potential aiding transferor. When politics encouraged everyone to distribute and produce electricity without centralized planning, the electric flow potential has become versatile. Nowadays, a transferor has to adapt to fluctuant potentials, developing strategies to switch between its operational modes according to the ongoing production and consumption.

A similar evolution has happened with Internet: knowledge is now produced and consumed by visitors to sites like Wikipedia. The rupture is caused by the change in knowledge potential, an important factor in the instantiation of the content transfer pattern.

Daily commuters are experiencing traffic jam in most cities. The storing capacity of the city is not the problem since all commuters finally find place and spend several hours in the city. The transfer content pattern underlines that the destination is hosting an incoming flow with a limited throughput.

Without the content transfer pattern these disruptive cases could not so easily be understood.

V. Conclusion

Usually, collecting architecture patterns from many sources is tedious and progress is fortuitous. In addition, best practices in pattern elicitation recommend that the interest in a pattern candidate be confirmed by many sources. The proposed method elaborates patterns from the systemic a priori Tables 1 and 2. It offers a complementary approach to mining them from experience. The illustrative field cases have been found by a systemic a priori and focused investigation. The variants with respect to flow potential emerged from the field cases. It was not implied by the a priori pattern.

The resulting transfer content pattern gained in applicability scope by the very way it was found. Based on the systemic a priori tables, new patterns (like transforming container: cooking recipes, chemical reactors) can be methodically searched for.

Furthermore, it can be argued that a truly systemic approach starts with a finality from which everything else is derived, model included. The systems architect's action is led by his intent to identify new patterns from the presented a priori framework.

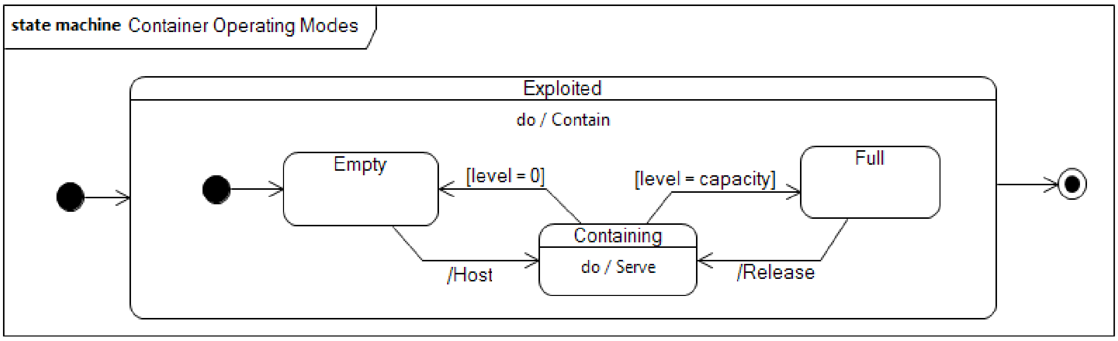

Figure 4. Operating Modes of a Container.

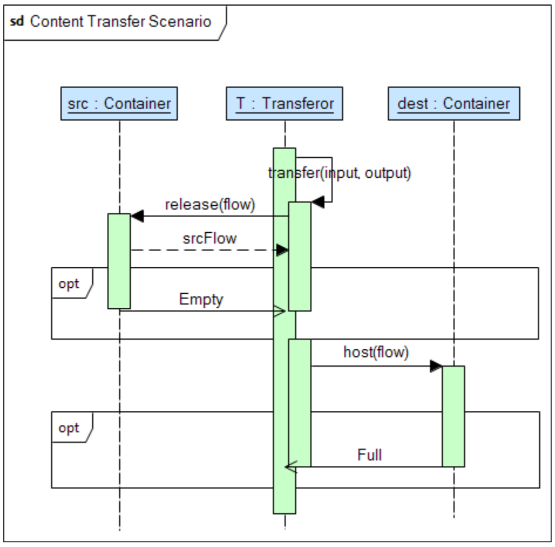

Figure 5. Scenario of Content Transfer.

TABLE III. CONTENT TRANSFER PATTERN

Pattern Name	Content Transfer
Aliases	None yet
Keywords	Container, content, transfer, flow potential
Problem Context	Most industrial processes include some transfer from one step to another. The nature of the content to be transferred is diverse, with transferring means adapted to the adverse, neutral, or aiding flow potential.
Problem Description	The way some content is transferred from one container to another depends on the flow potential and the means of transfer.
Forces	Transferor names are strongly influenced by disciplines and flow potential. The flow control may require processing algorithms specific to the application case.
Pattern Solution	The services provided by the containers are explicited. The category of transferor depends on the flow potential. Exceptional cases are identified.
Model	See figures in Section 4.
Interfaces	The mandatory flow is the content released from the source to be hosted by the destination through the transferor. Additionnal information about the levels of containers as well as control to the transferor may appear. The connections to the containers may be permanent.
Resulting Context	Depending on the flow potential and the means of transfer, additional flows (like energy or anchoring support to a pump) and components (like pipes) are required.
Example	Gas filling into a car at the gasoil station.The flow potential is adverse, so the transferor is a pump. The source container is the station cistern, the destination is the car tank. The flow of gas (matter) is streaming.
Known Uses	See illustratative field cases in Section 3
Related Patterns	None yet
References	This paper.
Pattern Rationale	The pattern is found from a systemic a priori candidate pattern.
Author	Laurent Gasser

ACKNOWLEDGMENT

We thank IRT-SystemX for approving the publication of this personal research. We also thank the reviewers whose constructive comments helped to improve the presentation of this method.

REFERENCES

[1] J. R. Cloutier and D. Verma, “Applying the Concept of Patterns to Systems Architecture”, Journal of Systems Engineering, vol. 10, no. 2, 2007, pp. 138–154.

[2] C. Haskins, “Using Patterns to Transition Systems Engineering from a Technological to Social Context”, Journal of Systems Engineering, vol. 11, no. 2, 2011, pp. 147–155.

[3] F. Pfister, V. Chapurlat, M. Huchard, C. Nebut, and J.-L. Wippler, “A Proposed Meta-Model for Formalizing Systems Engineering Knowledge, Based on Functional Architectural Patterns”, Journal of Systems Engineering, vol. 15, no. 3, 2012, pp. 321–332.

[4] T. Stålhane, O. Daramola, and V. Katta, “Pattern in Safety Analysis”, The Fourth International Conferences on Pervasive Patterns and Applications (PATTERNS 2012) IARIA, July 2011, pp. 7-10, ISBN: 978-1-61208-221-9.

[5] A. Hatchuel and B. Weil, “A new approach of innovative design: an introduction to C-K theory”, ICED’03, Stockholm, Sweden 14, 2003, pp. 109-124.

[6] E. Neufert and P. Neufert, “Bauentwurfslehre, 37. Auflage“, Friedr. Vierweg & Sohn, 2002, ISBN: 3-528-98651-4

[7] M. Gressmann, “Fachkunde Fahrradtechnik, 3. Auflage“, Europa-Lehrmittel, 2009, ISBN: 978-3-8085-2293-6

[8] G. Buchanan, “Shaum’s Outline of Theory and Problems of Finite Element Analysis“, McGraw-Hill, 1995, ISBN: 0-07-008714-8

[9] J.-L. Le Moigne, “La théorie du système général (4ème éd.)“, Presses Universitaires de France, 1994, (first edition 1977, at http://www.mcxapc.org/inserts/ouvrages/0609tsgtm.pdf, retrieved: January, 2015).

[10] Biomérieux, Vidia User’s Guides ref 38087-B-2006/07, 2006.

[11] J. W. Böse (ed) and B. Brinkmann, “Handbook of Terminal Planning”, Operations Research/Computer Science Interfaces Series 49, Springer, 2011, ISBN: 978-1-4419-8408-1_2

[12] E. Zehendner, “Operations management at container terminals using advanced information technologies”, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2013, NNT: 2013EMSE0714.

[13] Presse Polytechnique Romande, “Guide de la technique 3, l’énergie”; PPR, 1993, ISBN: 2-88074-221-8

[14] International Energy Agency, “Technology roadmap, Solar therma electricity, 2014 edition”, IEA

[15] H. Cruickshank, M.-P. Howarth, S. Iyengar, and Z. Sun, “A comparison between satellite DVB conditional access and secure IP multicast”, 14th IST Mobile and Wireless Communications Summit, 2005, pp. 19–22.

[16] A. Dearle, “Software deployment, past, present and future”, in 2007 Future of Software Engineering, IEEE Computer Society, 2007, pp. 269–284.

[17] OMG, “OMG Systems Modeling Language, version 1.3”, Object Management Group, June 2012.