ANALIZA SYSTEMOWA I ZARZĄDZANIE

Książka jubileuszowa
z okazji
50-lecia pracy naukowej

ROMANA KULIKOWSKIEGO

ISBN 83-85847-34-0

Druk: "ARGRAF" Agencja Poligraficzno-Wydawnicza, Warszawa
Skład: Barbara Kotuszewska

DEVELOPMENT CONTROLLED BY THE FLOW OF MESSAGES

Stefan Węgrzyn

Institute for Theoretical and Applied Computer Science
of the Polish Academy of Sciences, Gliwice

Developmental systems, classically characterised by central control according to a program expressed in the form of a generating word, can be looked upon from the viewpoint of distributed control by the flow of messages inside and between the elements according to a development algorithm. A canonical form of the letter is given. Examples are given of complex developmental systems consisting of an embryonic centre and patterns, possibly also subpatterns.

1. Introduction

A *developmental system* consists of elements that change in time according to a certain program. More precisely a developmental system is defined by (Węgrzyn *et al.* 1982):

a) a set Z of n categories of elements

$$Z = a_i \qquad i \in [1, n]$$

b) the set E of the operations that act on the elements

$$E = O_i(a_i) \qquad i \in [1, n]$$

c) an initial element

The ordered set of operations, which controls the evolution, will be called *development algorithm* (Bestougelf *et al.*, 1975).

The concept of a developmental system was first introduced by A. Lindenmayer (Lindenmayer, 1968; Herman *et al.*, 1975; Lindenmayer *et al.*, 1976). Since then the subject has given rise to an important body of literature. In the present author's previous publications (Węgrzyn *et al.*, 1990; Gille *et al.*, 1993) the elements of developmental system were assumed to be *synchronously* acted upon by operations controlled by a central "*generating word*".

In this paper a model is presented in which the development algorithm is no longer elaborated in a control centre but is created by the *flow* (or *stream*) *of messages*, within and between the elements of the system.

2. The operations

According to the definition a developmental system is made up of elements which undergo transformations.

The latter will be assumed to satisfy the following two conditions:

a) with the only exception of initial element, each element of a system has one antecedent already present in the system;
b) when acted upon by an operation an element can give rise to one or two elements.

These two assumptions are two fundamental axioms of biological system, which are developmental system whose elements are *cells*.

For instance operations on element a may be

$$O(a) \to b \tag{1}$$

$$O(a) \to bc \tag{2}$$

Where O is the operation and a, b, c are elements.

This means that

1) operation O acting upon element a yields element b,
2) operation O acting upon element a yields two elements b and c.

In that follows five types of operations will be considered. They are:

1) *Linear*, or *rectilinear*, *generation* L:

$$L(a) = ba \qquad \text{i.e.} \qquad a \to ba$$

2) *Rotative generation*, or *generation with change of direction* R:

$$R(a) = b(a) \qquad \text{i.e.} \qquad a \to b(a)$$

where the parentheses denote the change of direction (which may be specified in direction and magnitude by a subscript, e.g. $R_{\pm 45''}$)

3) *Bifurcation B*:

$$B(a) = bc \qquad \text{i.e.} \qquad a \to bc$$

(same remark as above for R)

4) *Bifurcation with change of direction C*:

$$C(a) = b(c) \qquad \text{i.e.} \qquad a \to b(c)$$

5) *Transformation*, or *differentiation T*:

$$T(a) = b \qquad \text{i.e.} \qquad a \to b$$

3. The development algorithm

According to the definition given in Par.1 the development algorithm of a system is the ordered set of operations which controls the evolution of the system starting from its initial element.

This can be looked upon from two viewpoints:

a) the development algorithm consists in specifying what operations are to act upon given elements;
b) the development algorithm consists in specifying what elements are to be acted upon by given operations.

This duality can be visualised in the form of matrix. It is shown in Figure 1 that the operations *B, T, C*, are applied to the elements a, b, c respectively. According to the viewpoint (a): operation B should act upon element a, operation T upon element b, operation C upon element c (columns of the matrix). According to the viewpoint (b): element a should be acted upon by operation , element c by operation C (rows of the matrix).

		Elements		
		a	b	c
Operations	*B*	x		
	T		x	
	C			x

Figure 1: Example of relations between elements (a, b, c) and operations (B, T, C) in development algorithm: matrix representation.

Hence two possibilities exist for the implementation:

a) *central control* of the operations which act on the elements of the system (Fig.2),

c) *distributed control* i.e., local control *by the flow of messages* (concerning data and arguments) inside and between the elements of the system (Fig. 3) (Węgrzyn et al., 1994).

In the latter case there is no centralised control: each element has its own local control centre (*LC*) which controls what operation should be performed on the basis of the messages (inner messages, and messages from the other elements of the system).

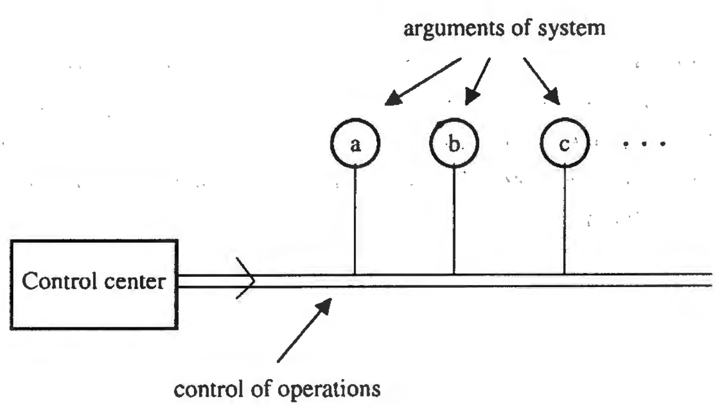

Figure 2: Central control

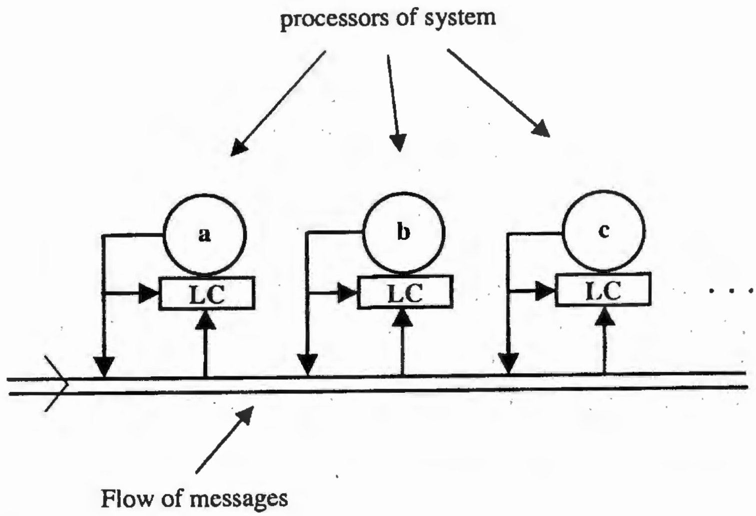

Figure 3: Distributed control by the flow, or steam, of messages. (Arrows are messages. LC means: local control centre.)

4. Canonical form

Two cases will be distinguished according to whether *L* and/or *R* operations are absent from or are present in the development algorithm.

4.1 Algorithm without any L, R operation

Consider the development algorithm

$$\left.\begin{array}{ll} B & a \rightarrow bc \\ T & b \rightarrow d \\ C & c \rightarrow (d)(d) \end{array}\right\} \qquad (3)$$

where:

a is the initial element,

b,c are the intermediate elements, and *d*   is the final element.

	a	b	c	d
a		x	x	
b				x
c				x
d				

It is seen on the matrix representation that the element *a* gives rise to *b* and *c* elements, and that *b* and *c* elements give rise to *d*. Note that the matrix is ***upper-tri-angular*** with ***zeros*** in the main diagonal. With central control the development consists of three stages after the initial state *a* (see Figure 4):

$$\underset{(B)}{a \longrightarrow} bc \underset{(T)}{\longrightarrow} dc \underset{(B)}{\longrightarrow} d(d)(d)$$

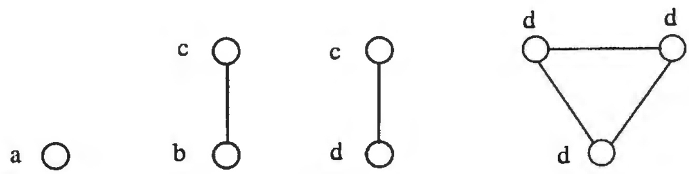

Figure 4: Development of system (3) with central control

If each operation requests T_0 seconds, the time necessary for the system to reach its final state is

$$T_f = 3T_0$$

In the case of control by the flow of messages operations *T(b)* and *C(a)* act in parallel (Raynal,1984; Roucairol *et al.*, 1985; Yonezawa *et al.*, 1987).
Hence the final state is reached after only two steps starting from the initial state a (see Figure 5):

$$a \xrightarrow{} cb \xrightarrow{\cdots} d(d)(d)$$
$$(B) \qquad (T \text{ and } C)$$

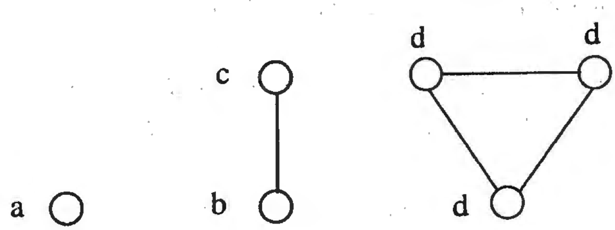

Figure 5: Development of system (3) with distributed control by the flow, or stream, of messages.

As a consequence the duration of the total development is

$$T'_f = 2T_0$$

i.e., the running time is shorter. The *speedup* produced

$$\frac{T_f}{T'_f} = \frac{3}{2} = 1.5$$

is a consequence of the presence of parallelism (Akl, 1989, pp.21-25; Vidal *et al.*1994).

4.2 Algorithm including L or R operations

Now consider the developmental system the algorithm of which is

$$\begin{array}{ll} L & d \rightarrow ed \\ C & e \rightarrow f(g) \end{array} \qquad (4)$$

or in matrix form

	d	e	f	g
d	x	x		
e			x	x
f				
g				

Note that the matrix is still ***upper-triangular*** but this time the main diagonal is not made up of zeros.

With central control the first stages of the development are (see Figure 6):

$$\begin{array}{l} d \rightarrow ed \rightarrow f(g)d \rightarrow f(g)ed \rightarrow f(g)f(g)d \rightarrow f(g)f(g)ed \\ \quad (L) \quad (C) \qquad (L) \qquad\quad (C) \qquad\qquad (L) \end{array}$$

With message flow control the first stages are (see Figure 7):

$$d \rightarrow ed \underset{(L\ and\ C)}{\longrightarrow} f(g)ed \underset{(L\ and\ C)}{\longrightarrow} f(g)f(g)ed$$

i.e., the growth has been accelerated by a factor 1.7 as a consequence of parallelism.

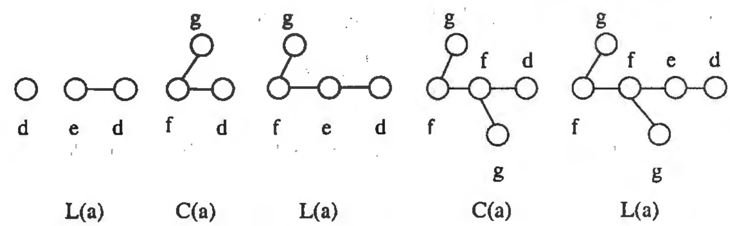

Figure 6: Development of system (4) with control center

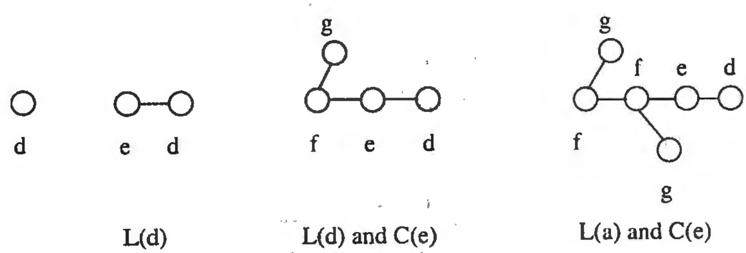

Figure 7: Development of system (4) with message flow

4.3 Comparison with the classical generating word representation

In our previous publications the theory of developmental systems was centred on the concept of a *generating word* made up of operations T, B, C, L, R, S synchronously acting at discrete instants of time (Węgrzyn *et al.*, 1982; Węgrzyn *et al.*, 1990, pp.12-17). The generating word can be expressed in the form of a graph or of a matrix. Two kinds of generating words have been distinguished: system without and with feedback. For the former the graph has the shape of binary tree and the matrix is upper-triangular; for the latter the graph possesses a loop and the matrix is no longer upper-triangular (Gille *et al.*, 1982, pp.380-382; Węgrzyn *et al.*, 1990, pp.16-18).

The *development algorithm* commented on here is an alternate expression of the generating word extrapolated for possibly nonsynchronous systems.

Note however that a development algorithm contains no *stagnation* operation S

$$S(a) = a \qquad \text{or} \qquad a \rightarrow a$$

as does any generating word. The reason is that an S "operation" simply means absence of an operation.

So far as matrices are concerned the matrix of a generating word is upper-triangular; its main diagonal is entirely made up of zeros if only T, B, C operations are present, but any L, R (or S) operation gives rise to a one. The same property has been observed for the operation *versus* element matrices considered in the foregoing two paragraphs, and can be shown to be true in general (Wyciślik, 1994). Only if the flow of the messages were not "laminar" (i.e. not constantly directed in the same direction) would the matrix cease being upper-triangular: this would be the equivalent of a generating word with feedback.

5. Complex developmental systems

A complex developmental system can often be broken down into two parts: an *embryonic centre* and the *patterns*. The role played by the former in the development consists in creating elements, which are the initial elements of the development of the different patterns of the system (Vidal *et al.*, 1991).

The development algorithm of an embryonic centre consists of B, C and T operations, *L* and *R* being excluded.

Consider for instance a complex developmental system with the development algorithm (see equation 3)

$$\begin{array}{ll} \mathrm{B} & a \rightarrow bc \\ \mathrm{T} & b \rightarrow d \\ \mathrm{C} & c \rightarrow (d)(d) \end{array} \qquad (5)$$

for the embryonic centre, and (see equation 4)

$$\begin{array}{ll} \mathrm{L} & d \rightarrow ed \\ \mathrm{C} & e \rightarrow f(g) \end{array} \qquad (6)$$

for the patterns. The matrices are respectively

	a	b	c	d
a		x	x	
b				x
c				x
d				

matrix of embryonic centre

	d	e	f	g
d	x	x		
e			x	x
f				
g				

matrix of patters

Note that both are upper-triangular and the first has only zeros in the main diagonal. The growth (Fig. 8) starts with the development of the embryonic centre, made up of three *d* elements (like in Fig. 5). Then each of these elements gives rise to a pattern which has the shape of an indefinitely growing branch with unit-length subbranches (Węgrzyn *et al.*, 1982, pp. 364-365; Węgrzyn *et al.*, 1990, pp. 6-7).

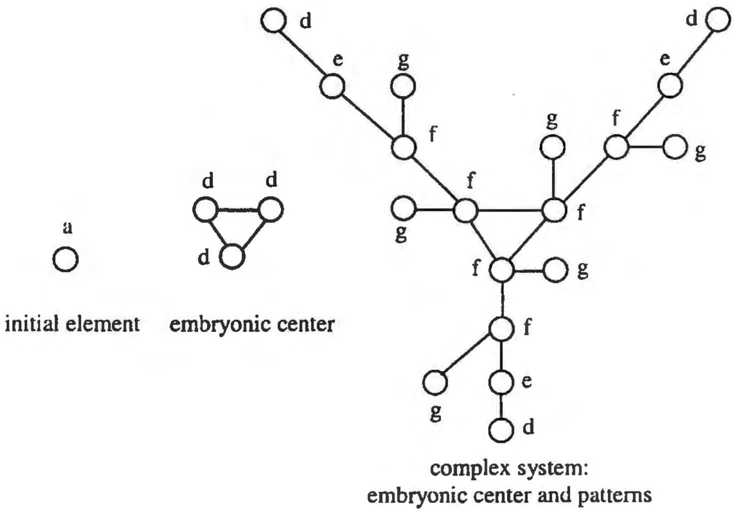

Figure 8: Growth of a developmental system consisting of an embryonic center (5) and patterns (6)

This can be compared to the first developmental stages of a living organism: the initial cell first divides into identical cells (*morula* stage), then the process of cell differentiation sets in and gives rise to the tissues and organs which will constitute the adult organism.

More complex developmental systems can be constructed which contain patterns and subpatterns as shown in Figure 9. An initial element gives rise to an embryonic centre which generates *patterns*. Then some elements of the pattern act in their turn as initial elements, each of which gives rise to an embryonic centre, which generates *subpatterns*.

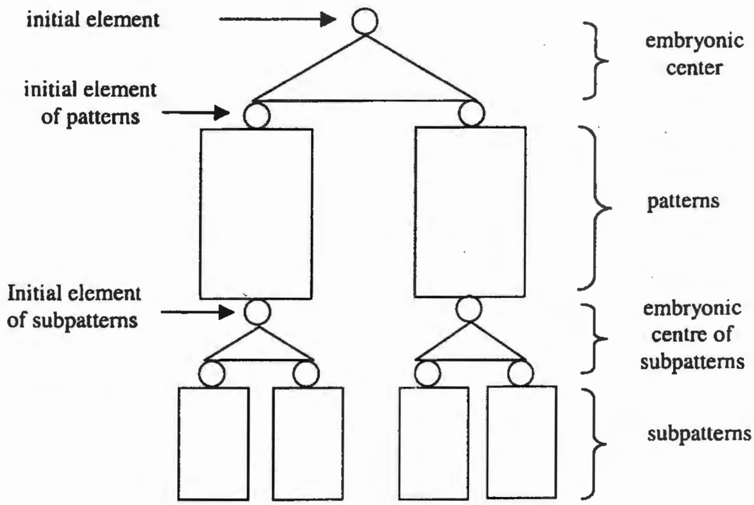

Figure 9: A complex developmental system with patters and subpatterns.

6. Example

The complex developmental system intended to illustrate the considerations of the foregoing Paragraph has the shape of a stem with branches and subbranches. At the first developmental level an initial element α gives rise to a rectilinear stem that bears *branches* of unit length, according to the algorithm

$$\begin{array}{ll} L & \alpha \rightarrow \beta\alpha \\ C & \beta \rightarrow \gamma(a) \end{array}$$

i.e., the growth occurs as (see equation 4 or 6)

$$\alpha \rightarrow \beta\alpha \rightarrow \gamma(a)\beta\alpha \rightarrow \gamma(a)\gamma(a)\beta\alpha \rightarrow \gamma(a)\gamma(a)\gamma(a)\beta\alpha \rightarrow \cdots$$

(See Figure 7 after having substituted α to d, β to e and a to g.)

At a second developmental level each at the a elements just generated behaves as the initial element of an embryonic centre, according to the algorithm (see equation 3 or 5 and Figure 4)

$$\begin{array}{ll} B & a \rightarrow bc \\ T & b \rightarrow d \\ C & c \rightarrow (d)(d) \end{array}$$

Finally each of these embryonic centres is at origin of a subpattern consisting of *subbranches*, according to (see equation 4 or 6 and Figure 7)

$$\begin{array}{ll} L & d \rightarrow ed \\ C & e \rightarrow f(g) \end{array}$$

The final result is shown in Figure 10.

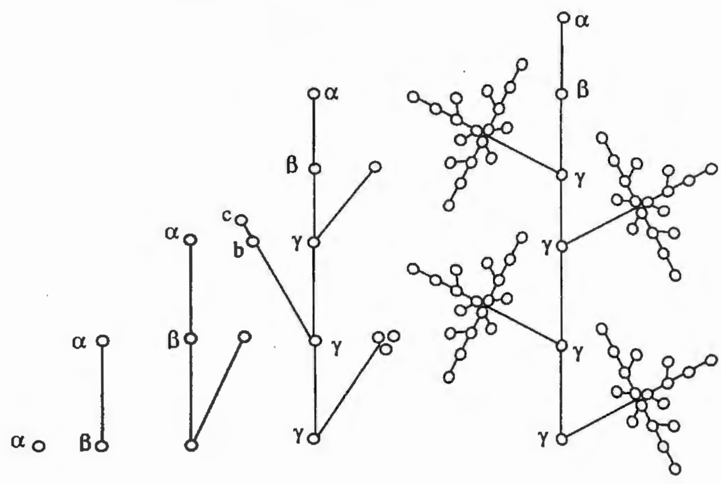

Figure 10: Example of complex developmental system with patterns and subpatterns

7. Conclusions

The article discusses distributed systems whose development is controlled by passing the messages between its elements (and a centre does not control it).

The development of such a system is always started by a single initialising element. The development of each particular pattern and the development of subpatterns within the patterns (and subsubpatterns in subpatterns) also begin from a single indicated element, called the initialising element.

Owing to such control system, the parallelism in realisation of patterns is possible and therefore the short of the whole system development time is assured.

ACKNOWLEDGMENT

The author wish to honour the memory of their late Colleague, Professor Jean-Charles Gille of University Laval, Quebec, Canada, whose participation was essential to the success of this research project.

References

1. Akl, S.G. (1989): *The Design and Analysis of Parallel Algorithms*, Englewood Cliffs, Prentice-Hall.
2. Bestougeff, H., Gulpin, C. and Jacques, M. (1975): *La Technique informatique,* Paris, Masson.
3. Gille, J.C., Vidal, P., Węgrzyn, S. Ouellet, Y. (1982): *Int. J. Systems Sci* ***13***, 379.
4. Gille, J.C., Vidal, P., Węgrzyn, S. (1993): *Int. J. Systems Sci.,* ***24***, 441.
5. Herman, G.T., Rozenberg, G., and Lindenmayer, A. (1975): *Developmental Systems and Languages,* Amsterdam, North-Holland.
6. Lindenmayer, A. (1968): *J. of Theoretical Biology*, **18**, 280.
7. Lindenmayer, A., Rozenberg, G. (eds) (1976): *Automata, Languages, Development; at the Crossroads of Biology, Mathematics and Computer Sciences,* Amsterdam, North-Holland.

8. Raynal, M. and Verjus, J.P. (1985): *Synchronisation de programmes parallèles*, Paris, Dunod.
9. Raynal, M. (1984): *Algorithmique du parallèlisme,* Paris, Dunod.
10. Suzuki, D.T., Griffiths, A.J.F. and Lewontin, R.C. (1981): *An Introduction to Genetic Analysis*, 2nd edition, San Francisco, Freeman.
11. Vidal, P., Gille, J.C., Węgrzyn, S., and Czachórski, T. (1994 or 1995): *Int. J. Systems Sci., **25** or **26**.*
12. Vidal, P., Węgrzyn, S. and Gille, J.C. (1991): *Bull. Polish Acad.-Sci. (Computer and Inf. Sci.), **39*** (4), 737.
13. Węgrzyn, S., Gille, J.C. and Vidal, P. (1982): *Int. J. Systems Sci., **13***, 359.
14. Węgrzyn, S., Gille, J.C. and Vidal, P. (1990): *Developmental Systems. At the Crossroads of System Theory, Computer Science, and Genetic Engineering,* New York, Springer.
15. Węgrzyn, S., Vidal, P., and Gille, J.C. (1994): *Bull. of Polish Acad. of Sci. (Tech. Sci.), **42**(1).*
16. Wyciślik, M. (1994): *Bull. of Polish Acad. of Sci (Computer and Inf. Sci.), **42***.
17. Yonezawa, A., and Tokoro, M. (editors), (1987): *Object-Oriented Concurrent Programming,* Cambridge, Mass, MIT Press.

ISBN 83-85847-34-0