The Utility of Pollination for Autonomic Computing

Holger Kasinger and Bernhard Bauer

University of Augsburg
Universitaetsstrasse 14
86135 Augsburg, Germany
kasinger|bauer@informatik.uni-augsburg.de

Abstract. From the biology's point of view, pollination is an important step in the reproduction of seed plants. From our point of view, pollination is a promising and novel, biological paradigm for future dependable and self-managing computing systems. This estimation is based on the characteristics the pollination process between plants and insects implies inherently.

To utilize pollination as a paradigm for self-managing and thus autonomic computing systems, this paper identifies the useful properties that emerge by the collaborative behavior of insects and plants during the pollination process. Based on this process the paper presents an artificial pollination system that implements these properties by adapting the natural architecture and behavior. Furthermore, the paper illustrates the practical value of this system by an application in aviation. Finally open issues and an outlook on future work are presented.

1 Introduction

Imagine sitting in an aircraft and looking out of the window whilst waiting for departure, you may see a buzz of activity: Dozens of baggage trains carrying innumerable pieces of luggage, catering trucks transporting fresh meals and drinks, service cars taking cabin crews to their aircrafts, or fueling vehicles pumping kerosene into the aircrafts' wings. In spite of this hectic overall picture, all activities seem to be intended and coordinated, what is the achievement of the ground control, a central facility at an airport. This institution is responsible for the coordination and management of all activities on the apron of an airport, in particular for every aircraft handling. In addition the ground control has to cope with any conceivable disturbances, e. g. absent ground vehicles, accidents on the apron, delayed or different typed aircrafts, unavailable passenger bridges, occupied ramps (the places for embarking and disembarking) due to delay, or other activities not finished properly.

However, in face of their valuable work, these centralized ground controls will become a bottleneck and single point of failure to airports in future, as the total passenger traffic world wide will continue its trend of the last decades and rise constantly. For example, Atlanta International Airport handled almost 76

millions of passengers in 2001, while in 2004 it have been almost 84 millions [1]. This results in an increased flight density at airports which causes the latter to expand in the same manner. This in turn boosts the management efforts of centralized ground controls more and more. Thus, ground control is clearly in need of new management approaches to cope with these future challenges.

Limiting management and administration efforts for computing systems also is the vision of Autonomic Computing (AC) [2]. Future autonomous computing systems are supposed to feature system-level self-managing capabilities, i. e. they ought to be self-configuring, self-optimizing, self-healing, and self-protecting (also referred to as self-* properties). But in spite of the many prospective approaches in various fields that delivered a couple of contributions to future autonomic computing systems in recent years (for an overview see [3]), this vision is not procurable easily. A remaining research challenge is that elements of an autonomic system have to share a set of common behaviors, interfaces and interaction patterns that are demonstrably capable of engendering system-level self-management [4].

To meet this challenge, it might be a good idea to throw a glance at nature. The adaptation of self-organizing biological systems [5] is a common method for the solution of artificial problems. For example, Swarm Intelligence [6] uses the collective behavior of biological systems (e. g. ant colony foraging, bird flocking, or termite mound construction) as paradigm for solving optimization problems. Also Autonomic Computing already makes use of biological paradigms in various fields (e. g. [7, 8, 9]), even the AC initiative [10] itself is based on a biological paradigm: the autonomous nervous system. Thus, looking for biological paradigms engendering autonomy at system-level and adapting their architecture and behavior will be a promising way for building future autonomous systems.

In this paper we present such a novel, biological paradigm: pollination of plants. From the biology's point of view [11], pollination is an important step in the reproduction of seed plants. Thereby pollen grains – the male gametes – are transfered from the anther of a flower to the carpel of a flower, i. e. the structure that contains the ovule – the female gamete. Pollination is not to be confused with fertilization, which it may precede. From our point of view, pollination can evolve into an important biological paradigm for future autonomic computing systems. This estimation is based on the self-* properties the pollination process between plants and insects implies inherently. Thereby system-level autonomy is not a result of one homogeneous but of two heterogeneous organizations, plants and insects.

The rest of the paper is organized at follows: Section 2 provides some background information on Autonomic Computing as well as on the biological pollination process. Section 3 identifies the emerging self-* properties of this process, that make pollination useful for AC, and presents an artificial pollination system that implements these properties by adaptation. In section 4 we illustrate the practical value of the artificial pollination system by applying it as a new management approach to the initial scenario presented above. Section 5 concludes and presents open issues as well as an outlook on future work.

2 Background

This section provides some needful background information on Autonomic Computing as well as on the natural pollination process between plants and insects.

2.1 Autonomic Computing

The Autonomic Computing initiative was founded by IBM in 2001. For its vision of self-managing systems, IBM proposes a reference architecture for autonomic computing systems [12], that consists of four levels: On the lowest level *managed resources (MR)* are located, e. g. HW/SW-components like servers, databases or business applications, together forming the entire IT infrastructure. So-called *touchpoints* on the next level provide a manageability interface for each MR – similar to an API – by mapping standard sensor and effector interfaces on the sensor and effector mechanisms of specific MRs, e. g. commands, configuration files, events or log files. The next higher level is composed of so-called *touchpoint autonomic managers (TAM)* directly collaborating with the MRs and managing them through their touchpoints.

Generally an *autonomic manager (AM)* implements an intelligent control loop (closed feedback loop) called *MAPE loop*. The latter is composed of the components *monitor* (collects, aggregates, filters and reports MR's details), *analyze* (correlates and models complex situations), *plan* (constructs actions needed to achieve goals) and *execute* (controls execution of a plan). Additionally, a knowledge component provides the data used by the four components, including policies, historical logs and metrics. Together with one or more MRs, an AM represents an *autonomic element (AE)*.

A TAM also provides a sensor and an effector to *orchestrating autonomic managers (OAM)* residing on top level. The latter are responsible for system-wide autonomic behavior, as TAMs are only responsible for an autonomic behavior of their controlled MRs.

2.2 The natural pollination process

In nature pollination involves different components and sub-process.

Pollination components Normally two components are involved in the pollination process, plants – more precisely the flowers of a plant as pollen source and pollen sink – and pollination vectors – agents carrying pollen from the source's anther to the sink's stigma (the receptive part of the carpel). Admittedly, there are a few plants that can self-pollinate, but as this results in inbreeding, most species rely on cross-pollination by some kind of pollination vector to accomplish pollination. The pollination vector not essentially has to be an insect or an animal, also wind and water come into operation. However, many plant species do not bank on random pollination by wind or water, thus insects and animals are the preferred pollinators of most species. In some cases, the evolutionary link between a species and its pollinator has become so tight that each is dependent on the other's efforts for its continued survival.

Attraction process During bloom the flowers of a plant need to attract pollinators that pick up and deliver pollen (grains) respectively to accomplish pollination. For the attraction, flowers provide certain attraction cues that might be visual or olfactory.

Showy petals or sepals with obvious shape, size, and color for the vectors' vision are important visual cues. Of course not every pollinator is attracted by the same colors, e. g. butterflies and birds are only attracted to red and yellow colors. Additionally, there might be color patterns (e. g. bull's eyes or nectar guides) that form a high-contrast exhibit to make the flowers stand out against a background of green foliage. Such cues assist a pollinator to "see" the flowers and in beginning to concentrate its visits only on those with the same certain colors. However, some vectors have limited visual capabilities but an extensive ability to find a flower by its fragrance. Thus, flowers produce volatile chemicals that diffuse and are carried by air movements through the environment. A vector that is able to recognize such a fragrance and fly up the concentration gradient, can easily find the next flower of a particular species. Flowers over time have evolved a wide array of fragrances which results in efficient pollinator attraction too. Again, different pollinators have different sensitivities to certain fragrances, e. g. flowers specialized in attracting flies are famous for their fetid aroma.

Not until a successful fertilization succeeds the pollination, a flower ceases to attract pollinators, as there is no need of further pollen grains.

Rewarding process Nevertheless, attracting pollinators is not fruitful on its own, as pollinators usually are intelligent enough to avoid the energy waste of behaviors that do not result in some kind of reward. Thus, a flower needs to reward an attracted pollination vector so that it will perceive the reward as a result of its visit. The vector's intelligence will then allow it to decide to visit similar flowers nearby to obtain additional rewards. This is the reason why vectors visit only one flower species on a trip.

While collecting its reward the vector unconsciously picks up and delivers pollen grains by its underside. Vectors collect rewards as long as they have had enough or they can not find anymore. This remarkable vector behavior ensures an effective pollination. The vector's reward can be either nectar, pollen, behavior, or some combination of these. *Nectar* is a carbohydrate rich droplet that is used as an energy source for vectors. Hummingbirds, for example, must consume vast quantities of nectar to continue their high-energy method of flight. Bees collect nectar and evaporate it down to make honey for winter supplies. The *pollen* itself contains protein, starch, oil, and other nutrients. It is far richer than nectar in vitamins and minerals too. For bees, the collection and consumption of pollen is critical as it is their basic protein supply. Fortunately, pollinators on this account are not very careful in cleaning off sticky pollen that cling to their bodies. *Behavior* can also be a reward that gets a repeat visit by a vector. The vector must like the experience while visiting and come back for more.

3 An artificial pollination system

3.1 Pollination process properties

Over the past millions of years plants and insects have evolved a natural, autonomous system that exhibits various useful properties for AC:

Self-configuration: The evolutionary link between a species and its pollinator is responsible for a seamless incorporation of new plants and pollination vectors. A plant is incorporated as soon as a linked vector scents its fragrance, while as soon as a vector scents a linked fragrance, it is incorporated itself.

Self-optimization: Vectors carrying pollen faster will collect more reward. In addition, flowers providing higher reward will be visited more often. Both speeds up the pollination process by different strategies within the components.

Self-healing: The loss of pollination vectors yields (to a certain extent) to no significant disturbance of the pollination process, as other pollinators will pick up and deliver pollen grains instead of. The reason is, that flowers produce pollen as long as they are fertilized (or their bloom is over before respectively).

Self-protection: Reward is only provided to vectors that pick up or deliver pollen during its visit. Flowers are that structured, that no intruders can receive any reward without picking up or delivering pollen as a trade-off.

Self-adaptation: A plant (species) not adapting its attracting and rewarding to the available pollination vectors over the long run will finally die out. Vice versa, a vector (species) not adapting its behavior to the specific characteristics of the available plants will become extinct either.

Self-organization: Pollination exhibits all required aspects [13] for a self-organizing system: It exposes an *increase in order* – evoked by the attracting and rewarding –, is *autonomous* – it has no external control –, is *adaptable and robust w.r.t. changes* – it has no single point of failure – and is *dynamical*.

3.2 System architecture

The adaptation of the natural architecture and behavior requires some premises necessary for an efficient exploitation of the above pollination process properties:

(1) A single artificial pollination system represents a finite, natural pollination environment, e. g. a grassland or a piece of forest. The representation of the entire nature as a huge, single, and closed pollination system would be absurd. (2) Sun, wind and rain come not into operation within the artificial pollination system, neither as pollination vectors nor as influencing quantities. Thus, pollination is based on "living" vectors only. (3) The attraction of artificial vectors is based on olfactory cues (fragrances) only, as volatile chemicals are representable, see pheromones [14] for example. Visual cues would be nonsensical in the scope of autonomic computing systems. (4) The rewarding of artificial vectors is based on nectar rewards only. It would be counterproductive if vectors are allowed to consume picked up pollen. Defining an exiting behavior for software components is absurd either.
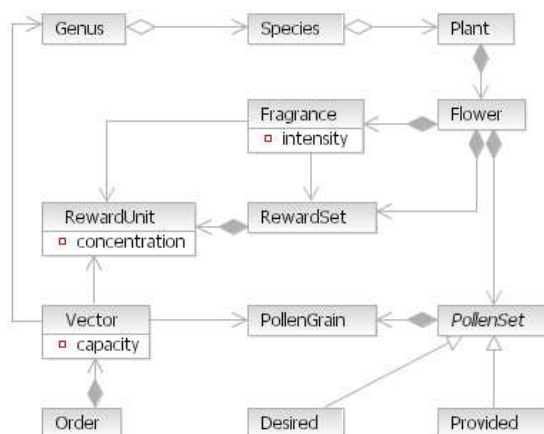
Fig. 1. Meta model of the artificial pollination system

Figure 1 depicts the meta model of the system architecture of the artificial pollination system (APS), that takes the above premises into consideration. **Plants** within the APS are defined by its **genera** and **species**. The scientific classification of natural plants into genera (e. g. roses or tulips) and species (e. g. Redleaf Rose, Gooseberry Rose, . . .) goes back to Linnaeus [15] and is adapted here as well. According to Linnaeus, a genus consists of one or more species, whereas a species consists of a plenty of entities (plants). A species may be sub-divided into subspecies, races, . . . , but this refinement is not mandatory here. Linnaeus regarded genera and species as disjunctive sets, what is reasonable for biology (a rose is a rose and not a rose and a tulip at the same time). But for the APS this disjunction would hold some disadvantages. Thus, without loss of generality, we admit a plant to be a member of one or more species, as well as a species to be a member of one or more genera at the same time – which represents evolution. As a consequence, a natural pollination environment can be considered as a special case of the APS. For zoology Linnaeus specified a scientific classification too. For the APS we only adapt the hierarchical term of order. In this case the natural disjunction remains, i. e. a **vector** belongs to exactly one **order** at the same time. Due to the fact that the system represents a finite pollination environment, the system boundaries are clearly defined and the number of plants and vectors is determinable at any time.

In contrast to the entire plant a **flower** is only allowed to be of a single genus and single species at the same time. From the fertilization's point of view the allocation of a flower to a species is not essential in either case. Sometimes it is sufficient, if the pollen grain a flower is pollinated with emanates from a flower of the same genus, independent of its species, and vice versa.

A **pollen grain** within a **pollen set** is of the same species and genus as the flower it is produced by. Note, a pollen grain includes no more information, in particular no hint on the flower serving as addresser or addressee for it.

A flower possesses a **provided** and a **desired** pollen set, each including a dynamically changing quantity of pollen grains. This diverges from nature in one aspect: A natural flower does not know about the quantity of provided and desired pollen grains. The reason for this divergence is the representation of fertilization, as the moment of fertilization is responsible for the cessation of attracting and rewarding vectors. The APS represents this moment by the time a flower provides and desires no more pollen grains. On this account the divergence has no effect on the overall process.

A **reward unit** within the **reward set** of a flower corresponds to a nectar drop. As experiments (see [16, 17]) have pointed out, that e. g. honey bees do not only determine a good food source by the quantity but also by the quality of the reward, more precisely by the sugar concentration of the nectar, the **concentration** is attributed to a reward unit. To cover the quantitative aspect, we additionally define the constraint that "*per picked up or delivered pollen grain a vector will receive one reward unit*". Thus, the size of the provided reward set of a flower is always equal to the current quantity of provided and desired pollen grains. Furthermore, a change of concentration affects all reward units within a reward set in the same way.

A **fragrance** propagates the current reward conditions of a flower and therefore consists of all the information vectors need to decide to visit the flower: The genus, the species, the reward concentration, and the quantity of reward units (for pick up as well as delivery of pollen grains) provided by the flower. Additionally, an **intensity** is attributed to a fragrance, what ensures two natural aspects: Firstly, the temporal volatility of a fragrance, and secondly, the route guide for a vector. Note, like in nature a fragrance consciously includes no information on the identity of the emitting flower. A vector follows a fragrance because it wants to receive an adequate reward, no matter from which flower of a certain species or genus. If the vector scents on its way to this flower another fragrance with better conditions, the vector may follow this new fragrance.

A vector is a pollinator for only one or a few genera and can only pick up or deliver pollen grains from flowers of these genera. This represents the natural fact that not all vectors serve as a pollinator for every genus, but only for elected ones – flies will not pollinate roses for example. Furthermore, a vector's **capacity** limits its ability to collect innumerable quantities of reward units (and pollen grains) – just like in nature. There, a bee, for example, that is full of nectar, has to fly back to its hive and deliver the collected nectar as honey before being able to collect further nectar. As hives are not directly part of the APS, we define the constraint that "*per delivered pollen grain a vector may consume two reward units*" to free its capacity again. This coerces a vector with no available capacity, first to deliver a few or all of its picked up pollen grains and to consume the respective amount of reward units, before picking up any further pollen grains. As the genus and species of the first collected reward unit of a vector predefine the only species to be visited on the trip (like in nature), the end of a trip is represented by the moment a vector has collected no more reward units. This is the time when all picked up pollen grains are delivered.

4 An autonomous aircraft handling system

To illustrate the utility of pollination for Autonomic Computing, we use the artificial pollination system for an autonomous aircraft handling system, that may help to reduce the ground control management efforts at airports in future. Therefore, consider the model depicted in figure 2, which represents an instance of the meta model in figure 1, and visualizes the mapping between APS elements and real aircraft handling entities at an airport.
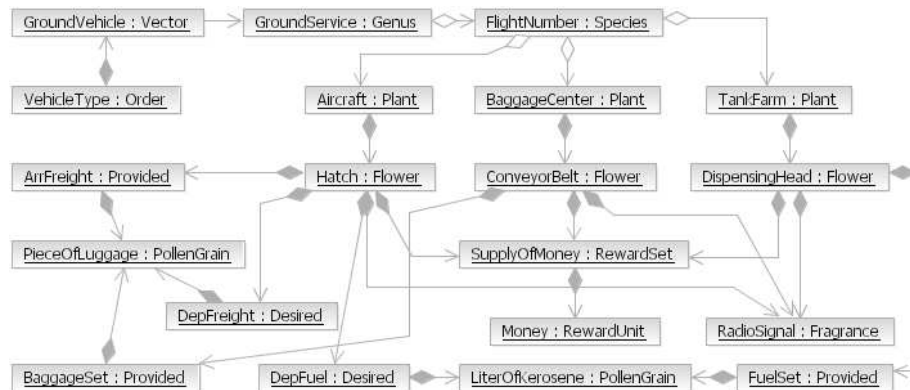

Fig. 2. Instance model of an autonomous aircraft handling system

To simplify matters, this instance model shows a flattened aircraft handling, where only two ground services, baggage handling and fueling, are required. Thereby each *ground service* is mapped on a genus. *Flight numbers* of arriving aircrafts, e. g. LH 457, AC 874, ..., each are mapped on a species, whereas the *aircrafts* themselves become a plant. Aircraft facilities, here the *baggage center* and the *tank farm*, are mapped on plants too. Flowers (*hatches*, *conveyor belts*, and *dispensing heads*) emit *radio signals* as fragrances, that attract *ground vehicles* (mapped on vectors) of a certain *vehicle type* (mapped on orders), e. g. here baggage trains or fueling vehicles. Just like in nature, such an attracted vector carries the *pieces of luggage* or *liters of kerosene* (mapped on pollen grains) from the provided pollen sets of flowers (here the freight by arrival *ArrFreight*, the *baggage set* and the *fuel set*) to the desired pollen sets (here the freight by departure *DepFreight* and the fuel by departure *DepFuel*) and hence are rewarded by the flowers with *money* (mapped on reward units) of the provided reward sets (mapped on the *supply of money*) of every flower.

Beneath this theoretical mapping, the AC reference architecture has to be applied, to make the autonomous aircraft handling work. Thus, every real entity (airport facilities, aircrafts, vehicles, ...) is represented as a managed resource and equipped with a touchpoint as a management interface. On top of these touchpoints autonomic managers, e. g. software agents, are placed, that assume

the corresponding role and behavior that are assigned to each resource by the mapping above. By these mappings and applications the aircraft handling proceeds in the same autonomous manner as the pollination process in nature and thus can make use of all the self-* properties identified in subsection 3.1, what may reduce the management efforts by ground controls.

5 Conclusion, open issues and outlook

This paper pointed out, that pollination is not only an important step in the reproduction of seed plants in nature, but also serves as a biological paradigm for future autonomic computing systems due to its properties implied inherently. This was accomplished by an adaptation of the natural pollination process between plants and insects as well as the corresponding sub-processes. The resultant artificial pollination system and its meta model respectively now enable the exploitation of these natural properties for the self-management of future systems, in particular the self-configuration, self-optimization, self-healing and self-protection of these systems. The future practical value of this paradigm was demonstrated by an example application for an autonomous aircraft handling system.

Nevertheless, the presented meta model provides no blueprint for all kinds of future autonomic computing systems. A domain-specific application requires a possible mapping of plants, flowers, fragrances, pollen grains and vectors on appropriate entities that are desired to run or perform a task autonomously. Beneath autonomous aircraft handling, one can think of autonomous manufacturing control, where robots (vectors) carry workpieces (pollen grains) to product machines (plants), or high rack warehouses with a similar behavior, for example. Of course these application scenarios already today run automatically, but not inevitably autonomously. By virtue of these versatile applications, a mid-term objective will be to expand the pollination system to an autonomic middleware for suchlike application domains.

However, this objective brings several open issues about. One issue is the management of the global system behavior by high-level policies. This requires knowledge about the correlations between the global system behavior and the local behavior of single components, in particular how to control emergence. Further issues are novel and enabling technologies supporting the intended APS behavior, like Semantic Web and Semantic Web Services, Grid, P2P, or multi-hop ad-hoc sensor networks.

Thus, the next step will be the implementation of an APS simulation. This may shed light on alterable system parameters, as the optimal relationship between the number of plants, flowers, and vectors, a flower's fragrance emitting frequency and the intensity (decrease) of a fragrance as well as the reward unit concentration adjustment. This enables an evaluation of the scalability, efficiency, robustness and low-latency of the APS and may help to meet some open issues.

References

1. Airports Council International: Passenger traffic 2000 – 2004 final. (Available at http://airports.org)
2. Kephart, J.O., Chess, D.M.: The vision of Autonomic Computing. Computer **36** (2003) 41–52
3. Sterritt, R.: Autonomic Computing. Innovations in Systems and Software Engineering **1** (2005) 79–88
4. Kephart, J.O.: Research Challenges of Autonomic Computing. In: 27th International Conference on Software Engineering. (2005) 15–22
5. Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in biological systems. Princeton Studies in Complexity. Princeton University Press (2001)
6. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From natural to artificial sytems. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press (1999)
7. Anthony, R.J.: Emergence: A paradigm for robust and scalable distributed applications. In: First International Conference on Autonomic Computing. (2004) 132–139
8. Birman, K.P., Guha, R.M.S.: Scalable, self-organizing technology for sensor networks. In: Advances in Pervasive Computing and Networking. Kluwer Academic Press (2004)
9. Saffre, F., Blok, H.R.: "SelfService", a theoretical protocol for autonomic distribution of services in P2P communities. In: First International Conference on Autonomic Computing. (2004) 326 – 327
10. IBM: Autonomic computing: IBM's perspective on the state of information technology. (Available at http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf)
11. Pollination: The Columbia Encyclopedia. 6th edn. Columbia University Press, New York (2001-04)
12. IBM: An architectural blueprint for autonomic computing. (Available at http://www-03.ibm.com/autonomic/pdfs/ACBP2_2004-10-04.pdf)
13. Wolf, T.D., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined. In: Engineering Self-Organising Systems. (2004) 1–15
14. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics **26** (1996) 29–41
15. Linnaeus, C.: Systema naturae per regna tria naturae, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis. 10th edn. L. Salvii, Stockholm (1758)
16. Camazine, S., Sneyd, J.: A model of collective nectar source selection by honey bees: self-organization through simple rules. Journal of Theoretical Biology **149** (1991) 547–571
17. Seeley, T.D., Camazine, S., Sneyd, J.: Collective decision-making in honey bees: how colonies choose among nectar sources. Behavioral Ecology and Sociobiology **28** (1991) 277–290