AUTONOMOUS UNITS TO MODEL COOPERATING LOGISTIC PROCESSES: BASIC FEATURES

Hans-Jörg Kreowski

University of Bremen
Department of Mathematics and Computer Science
kreo@informatik.uni-bremen.de
http://www.informatik.uni-bremen.de/theorie

ABSTRACT
In this short paper, the concept of autonomous units is introduced as a device to model logistic processes and their interaction and cooperation within a logistic network. The aim is to provide a general approach with rigorous semantics that allows the visual modelling of logistic processes in a systematic and structured way and supports the verification of required properties of the modelled systems.

INTRODUCTION
As the structural and dynamic complexity of logistic networks is increasing, central planning and control become more and more difficult and reach their limits. A possible alternative is the decentralization of planning and control, which requires some degree of autonomy of the components and processes of logistic networks. The aim of the Collaborative Research Centre CRC 637 *Autonomous Cooperating Logistic Processes - A Paradigm Shift and its Limitations* (see the webpage www.sfb637.uni-bremen.de for more details) is the investigation and application of autonomy as a new control paradigm for logistic processes. This short paper contributes to the interdisciplinary, long-term research project CRC 637 by introducing the concept of autonomous units as a device to model logistic processes and their interaction and cooperation within a logistic network or system in a formal way with rigorous semantics. The aim is to provide a general approach that allows the visual modelling of logistic processes in a systematic and structured way and supports the verification of required properties of the modelled systems. Moreover, the approach supports the paradigm of autonomy in that it allows all entities, components and processes of a logistic network or system to act and interact in a self-controlling manner to a certain degree.

An autonomous unit may be any component of a logistic system that plays a specific role, performs some actions or carries some information. In the particular context of transport logistics, it may be a vehicle, a container, a freight item, or an RFID tag. To form a logistic process, network or system, autonomous units exist and coexist in a common environment, where they may cooperate or compete. In the case of an intermodal transport problem, the environment consists of all relevant sites (may they be cities, ports, stations, airports, freight centers, etc.) and connections between them (may they be roads, railways, waterways, communication channels, etc.). Moreover, the environment contains the available transport vehicles and the goods to be transported as well as all the necessary information like quantities, destinations, departure and arrival times, tour plans, etc., so that it is meaningful to be changeable over the time. In other logistic contexts, the environment may be chosen accordingly as a supply chain, floor plan, etc. Autonomous units have capabilities to move and act in the environment, they have goals that they try to reach, and they can communicate with each other. The autonomous units run in a potentially non-deterministic way, meaning that whenever they perform a step, they can choose among various possibilities in general.

In: Palwar KS et al.(eds) Proceedings of the 10th International Symposium on Logistics (ISL 2005), pp 377-380, 2005.

There is no global control to choose the next action. Each unit controls itself autonomously to cut down the non-determinism. The actual choice of the next action depends on the type of autonomy each unit is provided with. Typically, the most promising action to reach the goal is chosen according to given priorities or local evaluations.

The concept of autonomous units generalizes graph transformation units as studied in Klempien-Hinrichs, Kreowski, Kuske, and Schürr (1997, 1999, 2004), and it is a rule-based instantiation of the idea of agents (see, e.g., Wooldridge & Jennings (1995, 2000)). It is intended to enhance the usefulness of modelling concepts in logistics like, e.g., Arns et al. (2002), Scholz-Reiter & Höhns (2003), and von Mevius & Pibernik (2004).

AUTONOMOUS UNITS

An autonomous unit has a goal, some capabilities and a control condition. The goal (if not informally formulated in natural language) may be some formula in a suitable logic to describe what should be achieved or become true. There are two types of capabilities. On one hand, a unit has got some rules to perform local changes on the environment. On the other hand, a unit can ask for help from the outside, delegating tasks to other units. Finally, control is any mechanism that allows to choose the next action in each situation.

More formally, an **autonomous unit** is a construct $unit = (g,U,P,c)$ where g is a **goal** (formulated in a proper logic or language), U is a set of identifiers naming **used autonomous units** (that are asked to help), P is a set of **rules**, and c is some **control condition**.

As, for example, the environment of a transport system may be a transport net with sites as nodes, connections between sites as edges, and further information attached as labels to nodes and edges, a rule is some facility to change the current environment locally, i.e. a transport net transformation rule that can add and remove nodes and edges and change labels if applied. In other logistic contexts, the rules specify corresponding local changes. In general, a unit is non-deterministic with respect to its rules and used units, meaning that several rules may be applicable and several helping units may be able to do their jobs in any current environment. The simplest and most trivial form of autonomy in this situation would be to choose the next action by chance. The control condition of the unit provides a more sophisticated form of autonomy. It describes how to cut down the non-determinism by forbidding some of the possible actions. Ideally, it will always choose a unique next step or terminate, but usually some non-deterministic choice will remain. There are two typical kinds of control conditions. The first kind determines the order in which rules are applied or helping units are called. For example, the regular expression $(x+y);z^*$ requires that either x or y be performed as the first step and then z arbitrarily often, where x, y, and z are rules or used units. Alternatively, one may replace the * by the exclamation mark !, which means that the iteration of z is performed as long as possible. The second kind of control conditions is based on priorities which are either given directly for the rules and used units or depend on some evaluation. The latter allows units to choose their next steps by taking into account some information available in the current environment and lessons learnt from earlier experiences. In this case, one can speak about autonomy in the proper sense. Finally, goals describe the results or effects units are desired to produce. Usually, they may be expressed in some logic or in some other suitable language.

For example, one may like to model the shipment of packages (being transport goods of a fixed form) by transport vehicles as a system of autonomous units with the emphasis on packages with some degree of autonomy. An autonomous unit that specifies a **vehicle** may provide tour planning rules on one hand and loading and unloading rules on the other hand. In particular, it can make shipment offers to packages. An autonomous unit that specifies a **package** may provide rules to place shipment requests (including information about origin, destination, time restrictions, etc.) to compare shipment offers, and to choose the best one among them. To achieve shipment, vehicles and packages must interact with each other in a proper way. There must be a shipment request of a package before a vehicle can make a shipment offer. And there must be some offers before a package can choose the best one. While the choice among various offers may involve some evaluation (that can be done by the package unit itself or can be delegated to some used units), the control condition *shipment_request*;**vehicles***;*choose_offer*! guarantees the desired order of events with respect to a package. The control condition **packages***;*shipment_offer** is part of a proper vehicle control. In both cases, the package units and the vehicle units use each other in the sketched negotiation. Finally, the goals should be mentioned. The goal of a package includes to reach its destination in time and for a low fare. The goal of a vehicle concerns full loads all the time, optimal tours, high net profit, and other logistic parameters one may think of. Although this example is necessarily quite simplified, it may give a first idea how autonomous units act and interact and model logistic processes in this way.

AUTONOMOUS PROCESS SEMANTICS

The semantics of a logistic process, network or system consisting of a family of autonomous units is given in terms of environment transformations that are composed of rule applications and activities of helping units chosen according to the control conditions. This operational semantics describes the simulation of the modelled processes and their cooperation, which can be visualized if the environments have got a visual representation. In addition, the operational semantics provides an induction schema to prove properties of the systems.

More formally, let *ENV* be the set of environments, let *unit* be one of the autonomous units of a given system *Net* of autonomous units, and let $CHANGE(unit) \subseteq ENV \times ENV$ be a binary relation on the environments describing changes that can take place in addition to the changes *unit* produces while it acts autonomously. Then a **computation** or **process run** of *unit* is a sequence of environments $E_1 \ldots E_k$ obtaining the input/output pair (E_1,E_k) such that (E_i,E_{i+1}) is obtained by the application of a rule of *unit* or the call of a used unit or belongs to *CHANGE(unit)* for $i=1, \ldots ,k-1$ and the control condition is satisfied. The set *SEM(unit)* of all obtained pairs is called **semantic relation** or **input/output relation** of *unit*. Accordingly, a **computation** or **process run** of the whole system is a sequence $E_1 \ldots E_k$ obtaining (E_1,E_k) if it is a computation of each of the units of the system. This means in particular that (E_i,E_{i+1}) is obtained by the application of a rule or the call of a used unit of some of the units of the system for $i=1, \ldots ,k-1$ so that this sequence describes the interaction of all units with each other within a single run of the system. The set *SEM(Net)* of all obtained pairs is called **semantic relation** or **input/output relation** of *Net.*

Note that the semantics definition induces a proof schema to verify properties of the semantic relations by induction on the length of computations. This may be used in particular to prove that the goals are reached (or not).

CONCLUSION

In this paper, the basic ideas and features of autonomous units have been introduced with the aim to provide a general approach for modelling logistic processes. To shed more light on the significance of this new framework, further research is necessary in at least four directions.

(1) Detailed case studies in transport and production logistics are needed to illustrate how each unit works autonomously to reach its goal and how autonomous units interact with each other.

(2) The concept of autonomous units should be compared with other modelling approaches used in logistics like business process models, Petri nets, UML diagrams, and swarm intelligence including ant colonies, for example.

(3) The operational semantics of autonomous units as defined above is purely sequential, interleaving the actions of all involved units. The semantics should be extended by parallelism and concurrency because in reality the logistic processes run simultaneously so that the assumption of one action at a time is not fully adequate.

(4) In theory, the semantics definition allows one to prove properties (for example that the goals are reached) by induction on the length of computations. In practice, one would need tool support for verification. Hence, the concept of autonomous units should be related either to higher-order logic to apply theorem provers like ISABELLE or to the framework of model checking.

REFERENCES

Arns M, Fischer M, Kemper P & Tepper C (2002) "Supply Chain Modelling and its Analytical Evaluation", Journal of the Operational Research Society 53, pages 885-894.

Klempien-Hinrichs R, Kreowski H-J & Kuske S (2004) "Typing of Graph Transformation Units", Proc. ICGT 2004 Second International Conference on Graph Transformation, Volume 3256 of Lecture Notes in Computer Science, Springer, pages 112-127.

Kreowski H-J & Kuske S (1999) "Graph Transformation Units with Interleaving Semantics", Formal Aspects of Computing 11, pages 690-723.

Kreowski H-J, Kuske S & Schürr A (1997) "Nested Graph Transformation Units", International Journal on Software Engineering and Knowledge Engineering 7, pages 479-502.

Scholz-Reiter B & Höhns H (2003) "Integrated Software Agents: Enabling Technology for Collaborative E-logistics and E-business", International Journal of Computer Integrated Manufacturing, pages 517-525.

von Mevius M & Pibernik P (2004) "Process Management in Supply Chains - A New Petri-Net Based Approach", Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS '04) - Track 3, 10 pages.

Wooldridge M & Jennings NR, editors (1995) "Intelligent Agents - Theories, Architectures, and Languages", Volume 890 of Lecture Notes in Artificial Intelligence, Springer.

Wooldridge M (2000) "Reasoning about Rational Agents", The MIT Press.