Towards a spatial language for run-time assessments in self-organizing systems

De Angelis, Francesco; Di Marzo Serugendo, Giovanna

Towards a spatial language for run-time assessments in self-organizing systems

Francesco Luca De Angelis
Institute of Services Science
University of Geneva
Carouge 1227 GE, Switzerland
Email: francesco.deangelis@unige.ch

Giovanna Di Marzo Serugendo
Institute of Services Science
University of Geneva
Carouge 1227 GE, Switzerland
Email: giovanna.dimarzo@unige.ch

***Abstract*—In this paper we define a spatial language used to verify global properties of self-organizing systems at run-time. The language can be used to assess spatial properties of system components to check desired global properties of the system against emergent global behaviors arising from local interactions among components. The spatial language extends a logic- chemical- based coordination model that we have recently proposed and the verification of spatial properties is performed in a distributed manner among the nodes of the system.**

I. INTRODUCTION

Formal assessment of emergent global behaviors and properties in self-organizing systems represents an hard task to achieve and an important issue to tackle to make them usable in several realistic scenarios. The engineering of self-organizing systems is hardly influenced by the selection of the coordination model to adopt, which usually directly affects the choice of the middlewares and platforms used in the development stages. This means that the verification technique adopted to assess the emergent global behaviors of the final system has to cope with the features of the selected coordination model. To tackle this problem we have recently proposed a chemical-based coordination model [1] based on the concept of Logic Fragments, which supports the three following properties.

► **Injection of ad-hoc chemical laws on-the-fly**: Logic Fragments combinate logic programs to define ad-hoc coordination laws that can be injected on-the-fly at run-time.

► **Design-time assessment**: by resorting to its logic formalization, the interactions enforced by a Logic Fragment, producing a new emergent global behavior in the system, can be then formally verified at design time.

► **Run-time assessment of local properties**: Given that Logic Fragments allow for the injection of ad-hoc chemical laws at run-time, they can be also used to reason about the state of system components. In several typologies of self-organizing systems sometimes it is also important to verify the current global state of the system at run-time: this happens for example when some components of the system have to perform collaborative actions unplanned at design-time or when a formal certification of a spatial pattern is required; in all these cases, to be implemented such tasks require some knowledge about the states of the other entities of the system.

In this paper we extend the Logic Fragment coordination model with the definition of a spatial language exploited to convey properties to be verified. This is done by defining a logic language for assessing local and global properties of the system and resorting to the logic inference processes endorsed by Logic Fragments.

II. RELATED WORK

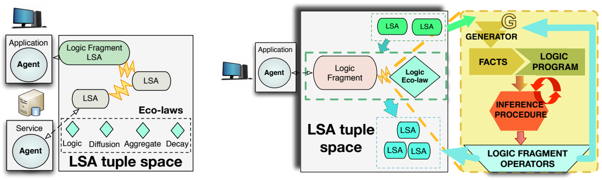
Query dissemination is a widely addressed problem in the sensor networks literature. The novelty of our approach is represented by the type of query disseminated over the network; in our model a query is a combination of logic programs and spatial statements: the former are active elements used to deduct information about the local states of nodes, the latter are logic formulae expressing global properties through relations of local ones, which are evaluated distributively. Consequently, the evaluation of logic sub-formulae can be implemented by resorting to the most effective dissemination algorithm presented in the sensor networks literature.

In this work we extend the Logic Fragment coordination model (Fig.1(a)), composed of the following components.

- **Live Semantic Annotations (LSAs)**: active tuples encapsulating information under the form of chemical solutions and grouped in two categories: (i) LSAs containing passive data and (ii) LSAs containing Logic Fragments.
- **LSA Space**: container of LSAs shared among all the agents of the same node. Within the LSA space, LSAs can interact among each others because of the presence of Eco-Laws.
- **Eco-Laws**: active components realizing chemical reactions, respectively for the evaluation of Logic Fragments and for diffusing, aggregating LSAs and for reducing information relevance over time.
- **Agents**: intermediary entities between the LSA Space and external applications (e.g. sensors, services). Every agent is associated with one LSA and it is invoked to implement specific actions when interactions among LSAs are fired.

Logic Fragments are combinations of logic programs and user-defined functions named generators used to infer new LSAs through the following logical process (Fig.1(b)):

- A subset of LSAs is converted into facts for logic programs by using generators.
- Facts are passed to logic programs and a logical inference procedure is executed.
- The inferred literals can be combined with the ones generated by other fragments by using Logic Fragment operators.
- The final inferred literals are then injected in the container under the form of LSAs.



(a) Logic Fragment coordination model.

(b) Evaluation of Logic Fragments.

Fig. 1

III. Spatial Language

Logic Fragments can be used to assess global behaviors of the system at run-time; the spatial language we propose resorts to them to automatize the reasoning process about topological system properties. The language we define is composed of spatial statements of the form $SS = (LF, SF)$, where LF is a Logic Fragment evaluated on the specific node executing SS and SF is a spatial formula of the same form of SS. The semantical meaning of a spatial statement is recursively defined by the semantical meaning of its SF component. The main predicate in spatial formulae is the dot ".", which verifies whether a literal is inferred in a Logic Fragment. For example the spatial statement $SS = (LF, Color(red).LF)$ is evaluated as true in a node N if and only if the literal $Color(red)$ has been inferred by the evaluation of the Logic Fragment LF on the node executing SS. A spatial formula SF can also contain inner spatial statements recursively composed with spatial operators, which distribute the evaluation of inner spatial statements as follows:

- $\forall_{neighs}(SS)$: $true$ if and only if the spatial statement SS is true in all the neighbors of N.
- $\exists_{neigh}(SS)$: $true$ if and only if the spatial statement SS is true at least in one neighbor of N.
- $\forall_{nodes}(SS)$: $true$ if and only if the spatial statement SS is true in all nodes of the network.
- $\exists_{node}(SS)$: $true$ if and only if the spatial statement SS is true at least in one node of the network.
- $\exists^k_{path}(SS)$: $true$ if and only if the spatial statement SS is true at least for all the nodes in a path of length k starting from N.
- $\forall^k_{paths}(SS)$: $true$ if and only if the spatial statement SS is true in all the nodes of all the paths of length k starting from N.

Spatial operators involve the execution of the SS component on the nodes that they identify; they are evaluated as follows:

- Given a spatial statement $SS = (LF, SF)$ created in a node N, LF is first evaluated on node by using the Logic Eco-Law. Then SF is evaluated; literals produced by the execution on LF can be used as parameters for SF.
- If SF contains only (eventually negated) formulae with "." predicates connected with logical operators $\{\wedge, \vee, \Rightarrow\}$ then SF is directly evaluated on N. The logic language involved in this step depends on the desired type of reasoning: for example particular distributed systems may need to manage partial and inconsistent information that usually arises self-organizing scenarios. This mean that spatial formulae can be evaluated by using logics ranging from the two-valued classical one up to multi-valued paraconsistent logics [2].
- If SF contains a spatial operator (e.g. $\forall_{paths}$) then the evaluation launches a distributed algorithm for evaluating the inner statement of the formula on to the nodes involved by the operator. The distributed algorithm used for evaluating the inner statement depends on the spatial operator and it involves the creation of flats or hierarchical structures (e.g. spanning trees) among the nodes of the network.
- When a statement SS is finally evaluated, the result is passed backwards either to the agent that injected the query or to the node that distributed the evaluation of SS (if SS is the inner part of a more complex spatial statement being evaluated). This step continues the evaluation of the remaining components of the spatial formula connected with logical operators $\{\wedge, \vee, \Rightarrow\}$.

A. Example

We imagine a network where node properties are represented by colors: we want to verify if all red nodes in the networks have at least one blue or green neighbor. This property is verified through the spatial statement in Eq. 1. The involved Logic Fragments are $LF = \emptyset$ (which produces no inferred literals) and $LF = lc$, which infers literals of the form $Color(X)$, where X is the color associated with the node executing it. Fig. 2 depicts the evolution of the computation of the spatial statement, which starting from node A and according to the involved spatial operators, distributes the evaluation of the inner spatial statements at first on node B and later on C and D (black arrows). In the final stage, the boolean values of the inner spatial formulae are combined bottom-up from C and D, aggregating values on B to produce the final value (T) on node A (red and green arrows).

$$SS \triangleq \Big(\emptyset, \underbrace{\underset{neighs}{\forall}(lc, \overbrace{\underbrace{Col(Red).lc}_{f_c} \Rightarrow \underbrace{\underset{neigh}{\exists}(lc, \overbrace{Col(Green).lc}^{f_e} \vee \overbrace{Col(Blu).lc}^{f_f})}_{f_d}}^{f_b})}_{f_a}\Big) \qquad (1)$$

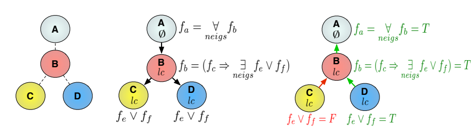


Fig. 2: Evaluation of the spatial statement in Eq.1.

IV. Conclusion and Future Work

In this paper we have defined a spatial language used to verify properties of distributed systems at run-time. The spatial language is based on the concept of Logic Fragments - combinations of logic programs, used to inject and assess ad-hoc chemical laws in a chemical-based coordination model for self-organizing systems. In future work we will focus on the distributed algorithms for verifying spatial statements at run-time in networks with mobile nodes.

References

[1] F. L. De Angelis and G. Di Marzo Serugendo, "Logic Fragments: a coordination model based on logic inference," ser. 17th IFIP WG 6.1 International Conference, COORDINATION 2015, Held as Part of the 10th International Federated Conference on Distributed Computing Techniques, DisCoTec 2015, vol. 90372015. Springer.

[2] A. Vitria, J. Maluszynski, and A. Szalas, "Modeling and reasoning with paraconsistent rough sets." *Fundam. Inform.*, vol. 97, no. 4, pp. 405–438, 2009.