Paper

Telemaco: A Language Oriented Tool for Graph-based Models Layout Optimization

Mauro Iacono and Stefano Marrone

Seconda Università degli Studi di Napoli, Caserta, Italy

Abstract—Progress of ICT is shifting the paradigm of systems organization towards a distributed approach, in which physical deployment of components influences the evaluation of systems properties. This contribution can be considered as a problem of graph layout optimization, well-known in literature where several approaches have been exploited in different application fields with different solving techniques. Then again, complex systems can be only studied by means of different formalisms which codification is the aim of language engineering. Telemaco is a tool that supports a novel approach for the application of graph layout optimizations to heterogeneous models, based on the OsMoSys framework and on the language engineering principles. It can cope with different graph-based formalisms by exploiting either their core graph nature or their different specialized features by means of language hierarchies. In this paper Telemaco is introduced together with its foundations and an example of application to Wireless Sensor Networks (WSN) deployment.

Keywords—graph optimization, modeling languages, wireless sensors networks, WSN deployment.

1. Introduction

Graph layout manipulation is a powerful tool that finds application in many different fields: from computer networks to mechanical modeling, from resources allocation to discrete events systems models, a graph structure appears to be inherent in the inner nature of problems. Optimizing graphs is thus a general solving approach that can exploit either common aspects or specialized issues of models. According to these issues, it is necessary to find an unified way to deal with such different models expressed in different sub-languages conform to graph based ones – language engineering is a discipline that best fits these needs.

In this paper the authors introduce Telemaco, an extensible tool for the optimization of graphs layout under customizable metrics. Telemaco is designed to transparently optimize graph-based models written according to user-defined modeling formalisms by exploiting the advantages of model engineering techniques. The description of models is based on a description framework in which each model is expressed in form of a given formalism. This framework allows formalisms to be designed as extensions of simpler formalisms, actually inheriting all the characteristics of ancestor formalisms, with which they stay fully compatible, applying a sort of inheritance concept with consequent advantages. While this approach allows a systematic development of models, it allows a generalization of some mechanisms, extending the reusability of tools to different kind of models without the need for a software rewriting. As well as a good designed languages hierarchy seamlessly enables Telemaco to correctly operate on new formalisms, its properly defined architecture allows it to be easily extended in order to embed new features in addition to or for a better specialization of the native ones.

Telemaco is part of the OsMoSys framework, both a methodology and a support environment for multiformalism models evaluation and analysis. OsMoSys offers a comprehensive and coherent support for model development and study through its family of languages for the definition of object oriented models and formalisms. In OsMoSys a model is composed by instances of formalism elements, which in turn are described by metaformalisms and that can be inherited from each other both at the formalism level and at the formalism element level, thus allowing of derived formalisms used to describe models that can exploit the advantages of base formalisms. This inheritance process allows Telemaco to define graph optimization primitives at any level of the formalisms hierarchy that can be automatically applied to any model described by any inherited formalism.

Telemaco can be used both in the general context of the OsMoSys Multisolution Framework (the support environment for the OsMoSys Multiformalism Methodology) or as a standalone tool. In this paper it is used standalone in order to focus on its characteristics and its architecture.

Together with modeling oriented languages, Telemaco also benefits of the query oriented languages offered by the OsMoSys framework. Telemaco implements specialized queries oriented to different graph metrics, according to the supported optimization methods and obtained as extension of the OsMoSys query languages.

In order to allow general and performing optimizations, the core metrics are implemented by genetic algorithms. This enables Telemaco to include several different optimality criteria, including heuristics.

Currently Telemaco is limited to layout optimization and only implements general graph optimization techniques. In the future, it will straightforwardly extended as soon as application to real world case studies will be analyzed in the next steps of this research activity. The application of genetic algorithms in Telemaco is not meant to be a comprehensive view of use of such technique but only wants to find a simple method to cope with different optimization heuristics, as different real world applications ask for.

The original contribution of Telemaco is in the ability of such tool, and of the underlying modeling approach, to cope with different aspects of system modeling and optimization. Since Telemaco essentially manipulates XML based models, it can easily be interfaced with the output of existing third-party tools and, since of its architecture, it can be easily extended to deal with a larger layout oriented set of problems.

After this introduction, a Section 2 gives a general introductions to relevant graph algorithms and genetic algorithms. Subsequently, model engineering is introduced in Section 3 with reference to the OsMoSys approach. Then the architecture of Telemaco is presented in Section 4, followed in Section 5 by a Wireless Sensor Networks based example and conclusions in Section 6.

2. Related Works

2.1. Graphs Optimization

Since graph optimization is a widely analyzed topic in literature, in this section the focus is limited to layout optimization. The problem has been solved with several different approaches and by different perspectives. Exact techniques are generally based on mechanical analogies while also many heuristic techniques proved to be effective.

From the first group Eades introduced the idea of considering springs in place of arcs to allow the optimization by using a mechanical potential function [1], further refined by Kamada and Kawai [2]. Particle physics inspired Fruchterman and Reingold [3], while Kumar and Fowler proposed a tridimensional version of the elastic method [4].

In the second group, Davidson and Harel proposed a heuristic function weighting vertex distribution, arc length and crossing and closeness to borders of the interest area [5]. Kirkpatrick, Gellat and Vecchi exploited simulated annealing [6], while Coleman and Parker [7] combined the advantages of [5] with the speed of [3]. Eloranta and Makinen introduce the use of genetic algorithms [8] and Branke and Bucher use a parallel algorithm based on the elastic approach in association with different criteria [9].

2.2. Tools for Graphs Optimization

Many tools available on the Internet exploit graphs optimization techniques in order to visualize information. A rough classification fitting the purposes of this paper refers to the implemented approach: physics based optimization or graphical optimization. In the first group, that uses algorithms simulating nodes as objects with masses and/or electric charges and consequently arrange the graph according to resulting forces, GraphOpt allows a layered structure to cope with very large graphs, CCVisu allows different energy models to tune the representation including a clustering function based on a LinLog approach, while GRINEdit allows plug-ins. In the second group, that rather implements graph structuring according to a selectable geometry (e.g., tree, circular, symmetric, hierarchical, orthogonal), GoVisual, based on the OGDF framework, allows animating layouts, GDToolkit allows optimization of parameters like length of the edges, the number of crosses and bends along them or the total area of the drawing on different graph types by exploiting object oriented features in the code and applying customizable layout constraints, while Guess applies graph manipulations and optimizations to database exploration and offers a sort of query language expressed in Gython. A list of references for these and other tools is available at http://www.dmoz.org.

2.3. Genetic Algorithms

Genetic algorithms are a heuristic method for search and optimization, inspired to the general principles of natural selection in biological evolution. The core concept is that an optimization process is designed as the creation of generations of candidate solutions, on which a fitness function is evaluated in order to detect the best candidates that are then combined by exchanging some of their characteristics to obtain the next generation. Genetic algorithms have been proved to best fit general situations where other methods can not use specialized knowledge about the heuristic function to optimize. An introduction to the topic can be found in [10]–[12].

3. A Language Oriented Approach

Model-driven Engineering (or Model Engineering tout court, ME) is a well known approach to the design and development of complex systems [13], that can be easily seen as a generalization of the widespread software engineering approach of the Object Managements Group Model Driven Architecture [14]. ME allows to separate conceptual aspects of design from implementation aspects, by exploiting the massive use of models and transformations between models [15]–[22]. These formal models capture different aspects of the design, including the model of the system architecture. Automatic transformations between different models, defined on the base of the underlying formalisms, allow designers to separately focus on parts of the problem and automatically adapt the results to the implementation. The goal of such approach is to obtain correctness-by-construction rather than construction-by-correction that is typical of several traditional and empirical system and software engineering approaches. Homogeneously defined models and metamodels form a Technological Space (TS) [23].

In order to support this philosophy of design, the infrastructure for models and manipulation is founded onto a coherent definition for description languages. A proper organization for languages consists of different levels of

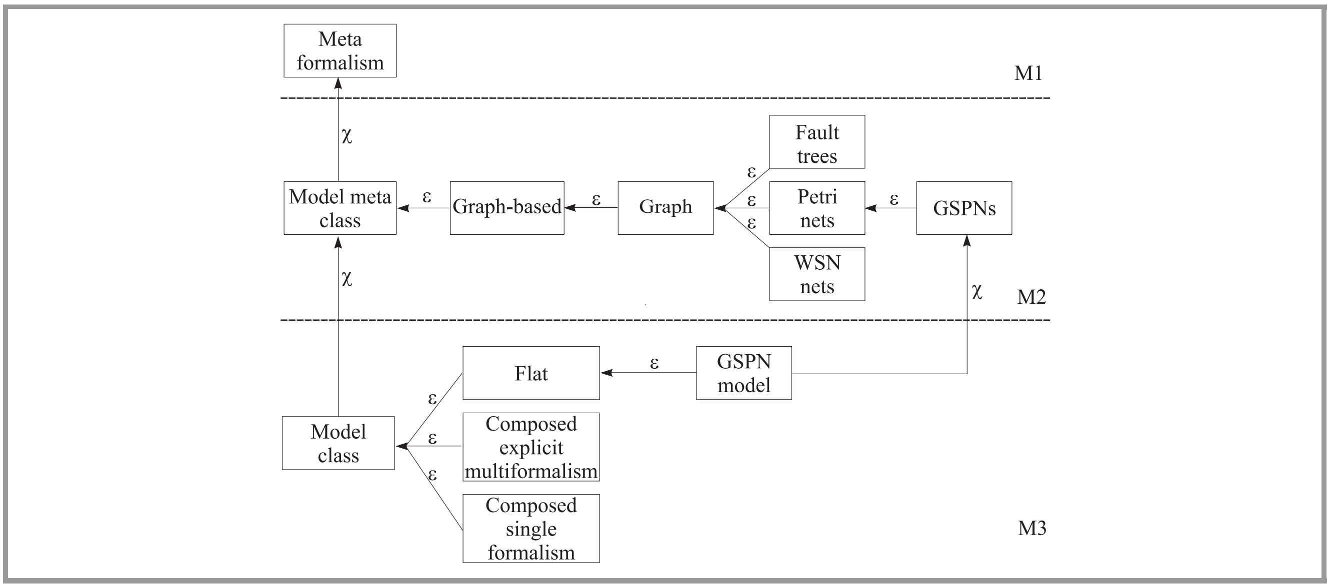

Fig. 1. Model Driven Engineering languages and models.

descriptions, that is the availability of a layer of languages aimed to describe models and a layer of languages aimed to describe such languages as is depicted in Fig. 1. A shared terminology defines as metamodels the languages aimed to models and as metametamodels the languages aimed to languages. The process of creating a models is thus considered as the instantiation of a number of elements on these layers, that generally define a tree structure through this stack. The motivation for this multilayered descriptions is twofold. The possibility of extending the number of available metamodels, by implementing new metamodels through existing metametamodels, and the possibility of establishing relations between models coherent with different metamodels by using existing relations between metametamodels. It is worth noting that the general representation of such models is usually a graph or can be mapped to a graph.

Other approaches exist that are founded on similar premises, but starting from a different point of view, such as the OsMoSys project, on which this paper focuses, and the SIMTHESys project [24]–[26]. The OsMoSys Multisolution Framework (OMF) [27] implements the OsMoSys Modeling Methodology (OMM) [28] that aims to multiformalism modeling, a modeling approach that allows different parts of a model (submodels) to be modelled with different metamodels (formalisms) in order to couple the description of each submodel with the best suited formalism. The OMM aims to build multiformalism models in order to evaluate some of their characteristics (i.e., performances, timeliness, dependability, availability) and supports the modeling process with proper semantic relations between submodels with different nature.

The OMF offers a number of model stacks (language families) to define not only the description of complex models, but also other relevant aspects as the queries with which the user can define the target of an analysis on the model or the kind of results a certain formalism can produce. Anyway, in this paper we will refer to the language family devoted to describe models. Models (model classes), metamodels (formalisms) and metametamodels (metaformalisms) are organized in order to allow the definition of model classes as compositions of submodels designed in different formalisms that can be related with each other by means of their description by a common metaformalism. New formalisms can be written from scratch by implementing their description in the most suitable metaformalism or can be obtained by extending existing formalisms with new elements or refining some elements, thus inheriting all the characteristics of the father formalism. This inheritance mechanism at the formalism level allows the modeler to exploit on new formalisms advantages designed for existing formalisms and automatically enables interactions between submodels not explicitly designed to interact with each others. For a deeper insight into inheritance in OsMoSys the reader can refer to [28].

The graph-based model description language family of OsMoSys can be considered in the ME perspective as a TS. This allows a further formalization of the process of generating any kind of graph-based (derived) model as in Fig. 1. In the figure, ε defines a is-a relation and χ defines a conform-to relation. On the model layer (M3) an example GSPN model, conform to the GSPNs formalism, is derived by a more abstract flat model (without submodels). On the formalism layer (M2), the GSPNs formalism is showed to be derived from the Petri Nets formalism, rather than from Fault Trees, another formal language, or WSN Nets, a description for Wireless Sensor Networks models. That in turn is derived from a simple Graph formalism, and next is derived by an abstract Graph-based formalism, that is a model metaclass (synonymous for formalism in OsMoSys). A model class conforms to a model metaclass, that in turn is conform to a metaformalism (M1).

Telemaco is designed to transform models in the OsMoSys graph-based TS and to be integrated into the OMF architecture as an OsMoSys adapter/solver couple [27]. Telemaco transforms a model written in a certain formalism into another model of the same formalism but with a different layout, according to a proper query formulated in the OsMoSys model query language. The tool is currently designed to operate on the graph-based formalism, and since of the OsMoSys languages inheritance properties, is capable of operating on all models conform to a formalism derived from it.
Since all languages of the OMF are implemented in XML, Telemaco can easily operate on the output of third-party tools, such as J-Sim, a a component-based, compositional simulation environment that has already demonstrated its effectiveness in modeling Wireless Sensor Networks [29].

4. Architecture of Telemaco

4.1. Layout Optimization

The problem of graph layout optimization, as seen, has been widely examined in literature. Two contributes affect the process: a functional aspect, connected to the nature of data represented by the graph, and an aesthetic aspect, common to all graphs. In order to detect unifying features in the process of graph optimization some layout and graph inputs and some heuristics are described. Inputs are:

- the dimensions of the visualization area; according to this parameter it is possible to scale properly all the computations;
- the initial positions for nodes;
- the arcs between nodes.

Possible metrics are:

- distance between nodes,
- uniformity: balances the density of the nodes in the available area,
- arcs intersection: avoids intersection of arcs if possible,
- average (or maximum) arcs length,
- symmetry.

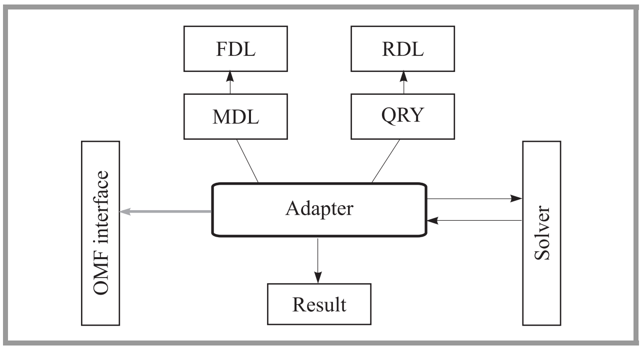

Fig. 2. An input/output view of Telemaco

The general structure of Telemaco is shown in Fig. 2. The solver (Telemaco.core) implements the optimization strategies while the adapter (Telemaco.ext) is in charge of accessing data from the model and the formalism (MDL, FDL) and from the query (QRY, RDL), and of producing results.

4.2. Solution Engine

The solution engine of Telemaco is based on genetic algorithms rather than physically derived algorithms because of the better potential of the first solution in terms of flexibility and extensibility. A genetic algorithm operates repeating a cycle of three phases until the desired number of generations has been reached, starting from an initial population automatically generated according to the initial parameters. The three phases are:

- **Selection**. During which the fitness function is evaluated on every element of the population to select which ones will contribute to the new generation. In order not to take always the locally optimal solutions, besides taking the best ones Telemaco implements two other strategies from the literature, namely the roulette [8] and the lottery strategies, the best strategies in a random subset;
- **Crossover**. During which it can happen that parts of the binary description of the coordinates of the nodes of two selected elements are swapped. Telemaco supports single and double swappings;
- **Mutation**. During which it can happen that some of the information about the coordinates of the nodes in a model are randomly changed. Telemaco supports four different mutation techniques.

Fitness evaluation is evaluated as the weighed sum of single fitness metrics that consider arcs intersections, node distances, density, angles between arcs, arcs lengths and arcs uniformity.

4.3. Internal Architecture

According to the OMF, Telemaco is composed by two components: an adapter and a solver (Fig. 3).

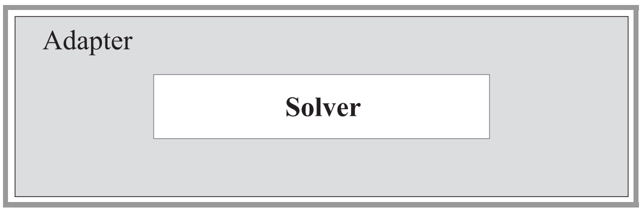

Fig. 3. Telemaco structure according to OsMoSys

In addition, Telemaco also offers a GUI (Telemaco.GUI) for the visualization of results when used standalone. The architecture of the tool is described in the UML class

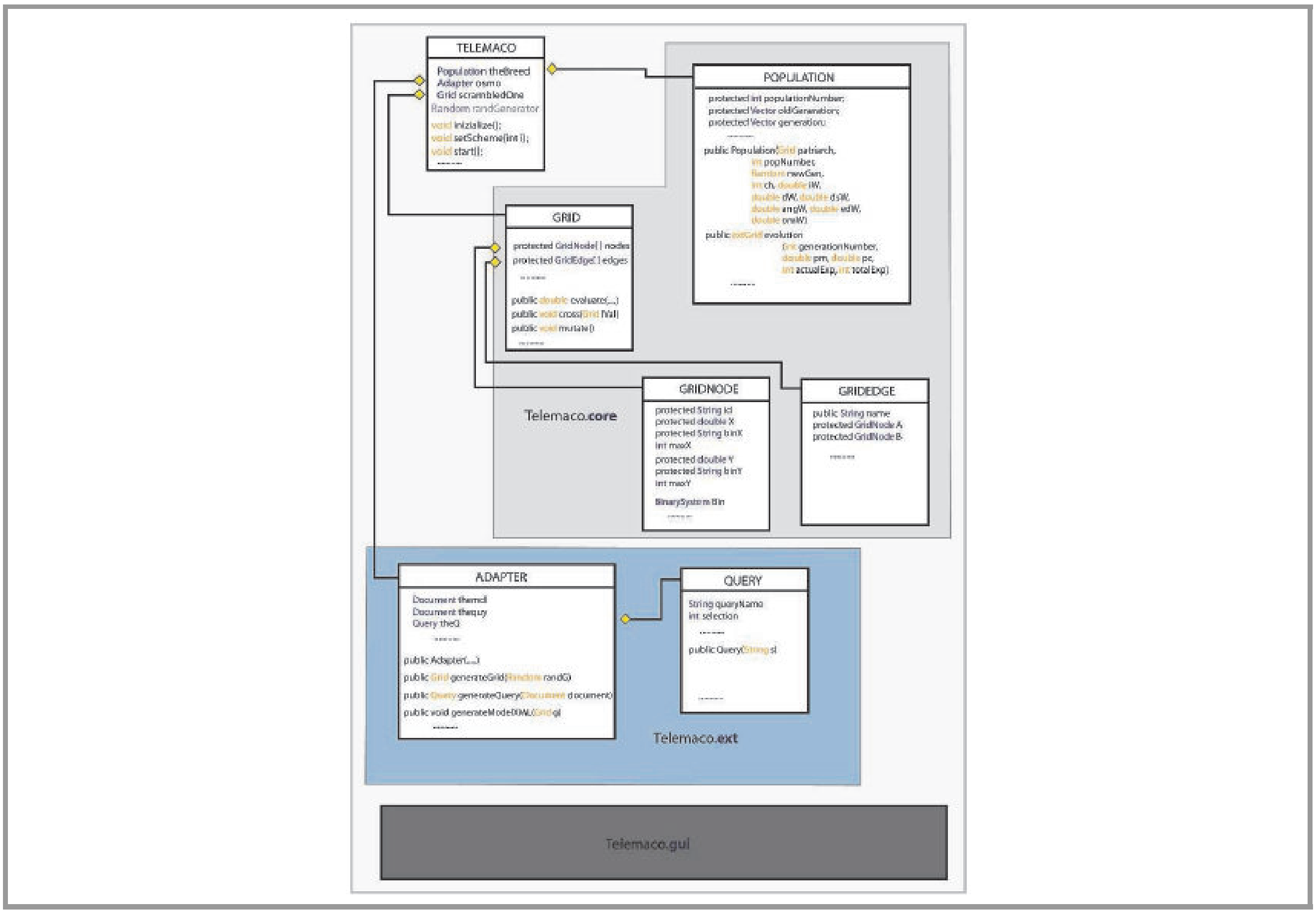

Fig. 4. A class diagram view.

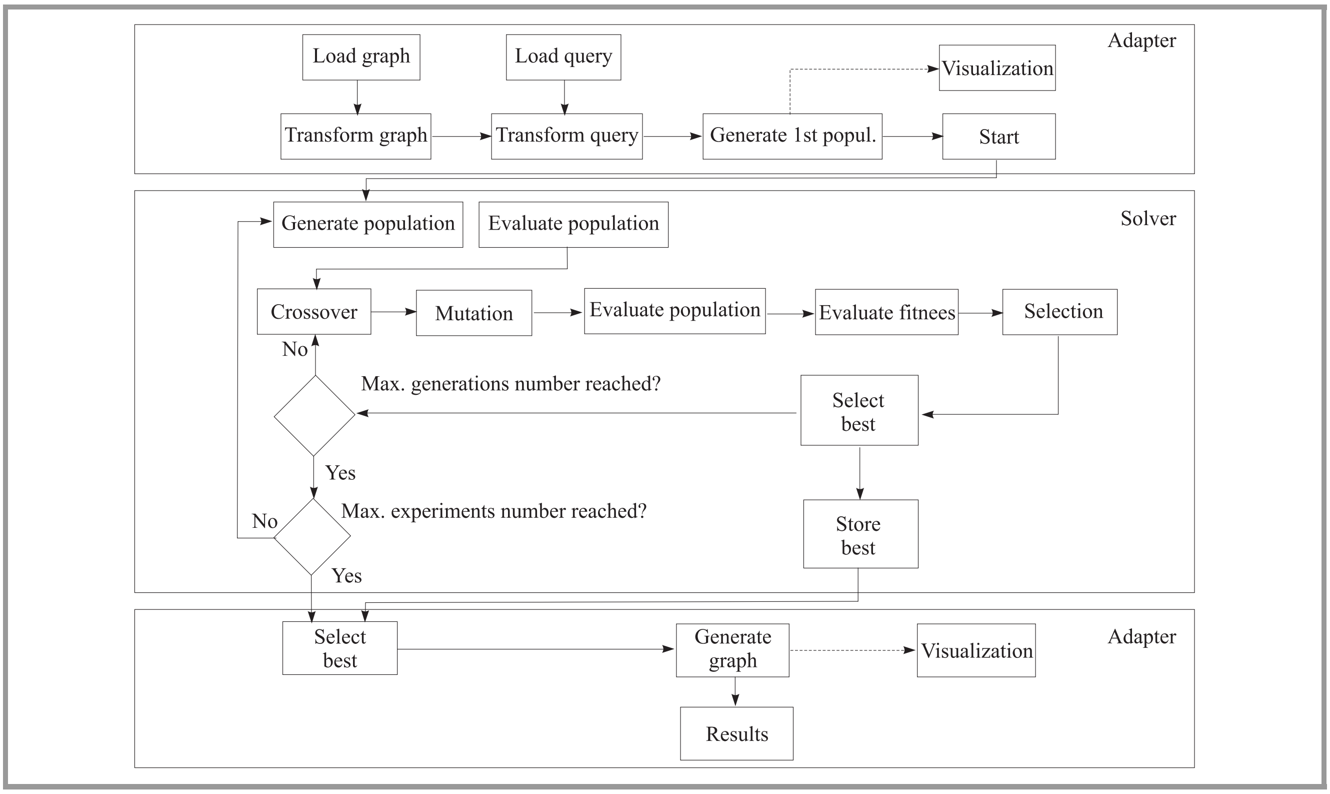

Fig. 5. Execution steps of Telemaco.

diagram in Fig. 4. Figure 5 presents a sketch of the operations performed by the tool.

4.4. An Application

In order to show how the tool works, we present here an example of the application of two queries, namely distinter and intersection, to the optimization of a graph representing a GSPN. The intersection query minimizes the intersection between arcs while the distinter query also aims to maximize distances between nodes. The initial situation is showed in Fig. 6 and both it and the query are described as follows:

```
<?xml version="1.0" encoding="UTF-8"
 standalone="yes"?>
<mdl type="FLAT">
  <GSPN fdl="GSPN.xml" name="Net2"
    Area="511">
    <Place name="P1" Tokens="2" X="10"
        Y="100"/>
    <Place name="P2" Tokens="0" X="20"
        Y="20"/>
    <Place name="P3" Tokens="0" X="300"
        Y="20"/>
<!--  TimedTransitions  -->
    <TimedTransition name="T1" X="30"
        Y="50"
     Rate="1.000000e+00" ServerType="0"/>
    <TimedTransition name="T2" X="80"
        Y="70"
     Rate="1.000000e+00" ServerType="0"/>
    <ImmediateTransition name="t3" X="25"
        Y="120"
     Weight="1.000000e+00" ServerType="1"
     Priority="2"/>
    <ImmediateTransition name="t4" X="30"
        Y="400"
     Weight="1.000000e+00" ServerType="1"
     Priority="2"/>
<!--  Arcs Section  -->
    <Arc name="Arc1" Weight="1" from="P1"
        to="T1"/>
    <Arc name="Arc2" Weight="1" from="T1"
        to="P2"/>
    <Arc name="Arc3" Weight="1" from="T2"
        to="P1"/>
    <Arc name="Arc4" Weight="1" from="P3"
        to="T1"/>
    <Arc name="Arc5" Weight="1" from="P2"
        to="T1"/>
    <Arc name="Arc6" Weight="1" from="P2"
        to="t3"/>
    <Arc name="Arc7" Weight="1" from="P1"
        to="t4"/>
  </GSPN>
</mdl>
```

```
<?xml version="1.0" encoding="UTF-8"
 standalone="no"?>
<mql rdlref="GSPN.rdl" mdlref="Nets.xml">
  <result name="distinter"/>
  <result name="intersection" />
</mql>
```

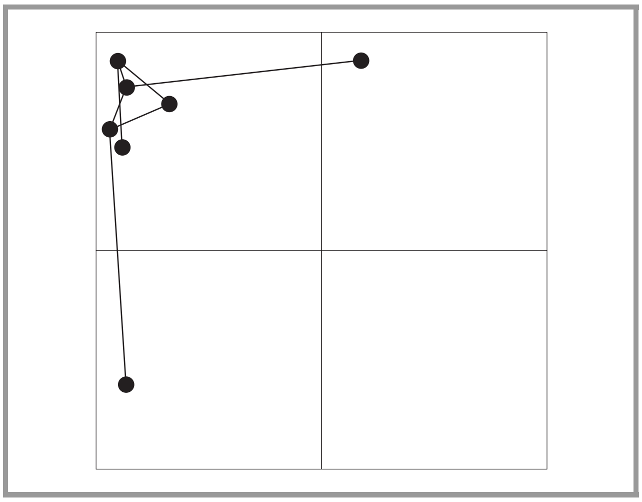

Fig. 6. Initial unoptimized graph.

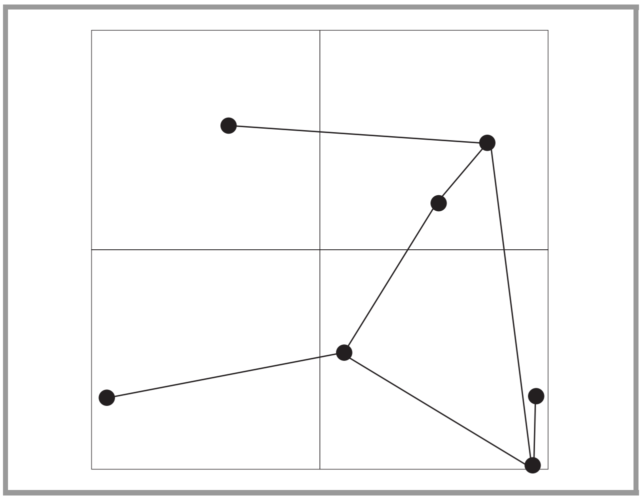

Fig. 7. Graph optimized according to distinter metric.

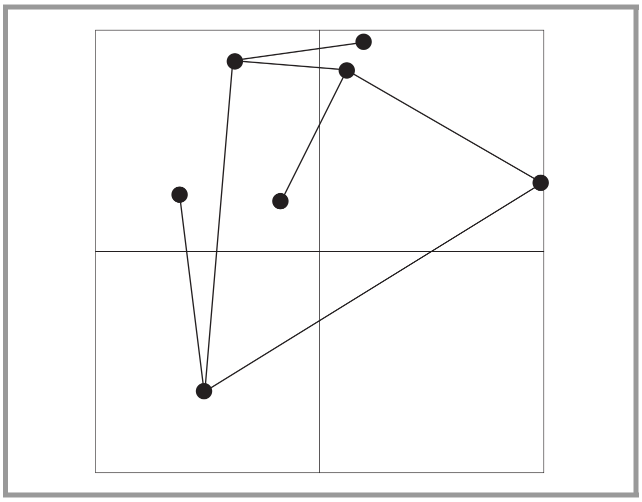

Fig. 8. Graph optimized according to intersection metric.

Figures 7 and 8 show the outputs of the two queries respectively after the execution of the genetic algorithm over few executions on several generations.

5. An Example – WSN Deployment

To demonstrate the use of Telemaco, a Wireless Sensor Networks (WSN) coverage example from the literature has been chosen. WSN are used to easily deploy sensors in areas of every dimension to collect data about the environment. In a WSN, sensors form a wireless ad-hoc network in order to vehiculate, with the lowest possible power consumption for transmissions, data towards some specialized nodes, known as High Energy Communication Nodes (HECN) and characterized by higher performances and connection to a communication channel towards the user. Optimal deployment of sensors is a well known problem [30] that depends on applications, and can be considered as the foundation of optimal dynamic relocation in Mobile Sensors Networks, see [31]–[34] for an introduction.

Our example is taken from [35], in which Jourdan and de Weck tackle a coverage problem for a military WSN with three different examples. Our example is a variation of the last of them, in which the WSN must cover at best an area by using sensors positioned so that no discontinuity can exist between their coverage. The area must be completely covered and at least one sensor must be connected to the HECN, which in turn has sensing capabilities. The authors design a multi-objective genetic algorithm approach specialized for this kind of applications. The presented example just aims to show the flexibility of Telemaco in facing problems for which it is not intentionally designed, by extending the tool for this purpose and comparing in general its results to the optimal solution obtained in the reference paper.

In order to capture the fact that a WSN is a specialized graph with two different node types (HECN nodes and sensor nodes) a proper formalism can be derived from the base Graph formalism. The new node types have a characteristic attribute, that is the coverage radius. Coverage radius will be described as an integer without loss of generality, and will be set to a default of 0 (non-working node). Each single sensor can have a different radius, in order to model heterogeneity of sensors or different sensing capabilities. The new WSN formalism will have three element types, two of which have been introduced and the third of which is the arc element, which actually represents the (fixed) connection that nodes will use to communicate. The two nodes will be specialized from Graph.Node (that is, the Node element of the Graph formalism) while the Arc will be derived from Graph.Arc.

The complete description of the formalism is as follows:

```
<?xml version="1.0" encoding="UTF-8"
 standalone="no"?>
<formalism parent="GRAPH" name="WSN"
 type="formalism">
  <elementType parent="" name="WSN">
    <elementType parent="GRAPH.Node"
        name="HECN">
      <propertyType name="Coverage"
            type="integer" default="0"/>
    </elementType>
    <elementType parent="GRAPH.Node"
        name="Sensor">
      <propertyType name="Coverage"
            type="integer" default="0"/>
    </elementType>
    <elementType parent="GRAPH.Arc"
        name="Arc">
     </elementType>
  </elementType>
</formalism>
```

The example is based on a WSN composed by a single HECN, with a bigger coverage radius, and five sensors, all with the same coverage radius. The arcs configurations is given and includes a loop containing the HECN in order to simulate redundant routing for better availability. The description of the WSN is as below:

```
<?xml version="1.0" encoding="UTF-8"
 standalone="yes"?>
<mdl type="FLAT">
  <WSN fdl="WSN.xml" name="WSN NET"
    Area="511" type="WSN">
    <Sensor name="P1" Coverage="100"
        X="10" Y="100"/>
    <Sensor name="P2" Coverage="100"
        X="20" Y="20"/>
    <Sensor name="P3" Coverage="100"
        X="300" Y="20"/>
    <Sensor name="P4" Coverage="100"
        X="500" Y="300"/>
    <Sensor name="P5" Coverage="100"
        X="350" Y="350"/>
    <!--  Energy Communication Node  -->
    <HECN name="T1" X="30" Y="50"
        Coverage="200"/>
    <!--  Arcs Section  -->
    <!-- Arcs -->
    <Arc name="Arc1" from="P1" to="T1"/>
    <Arc name="Arc2" from="P3" to="P2"/>
    <Arc name="Arc4" from="P3" to="T1"/>
    <Arc name="Arc5" from="P2" to="P1"/>
    <Arc name="Arc6" from="P3" to="P5"/>
    <Arc name="Arc7" from="T1" to="P4"/>
  </WSN>
</mdl>
```

Since the problem is about optimal coverage, the native metrics of Telemaco do not fit, being designed for the optimal visualization of graphs rather than to ensure the continuity of the covered area of sensors. The effects of native metrics is shown in Fig. 9 where the application of an intersection metric is presented. It is evident that the result is definitely inadequate in confront of one example layout (manually generated) depicted in Fig. 10.

In order to obtain a good result, Telemaco has been extended with two additional metrics. The first focuses on distributing the nodes at a distance that is as close as possible to the coverage radius. The second keeps the nodes

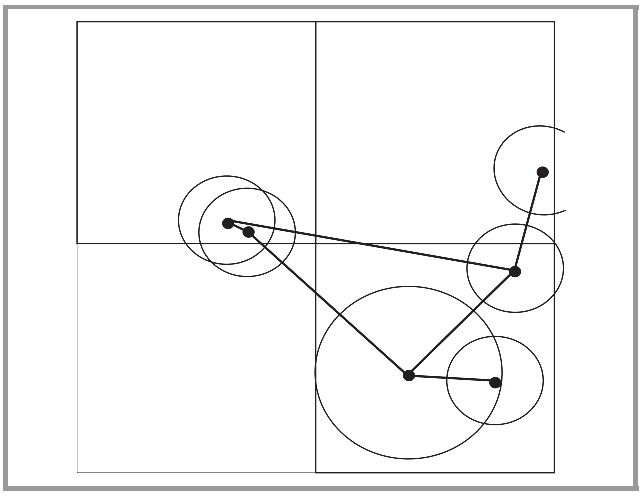

Fig. 9. WSN wrong optimization according to intersection metric.

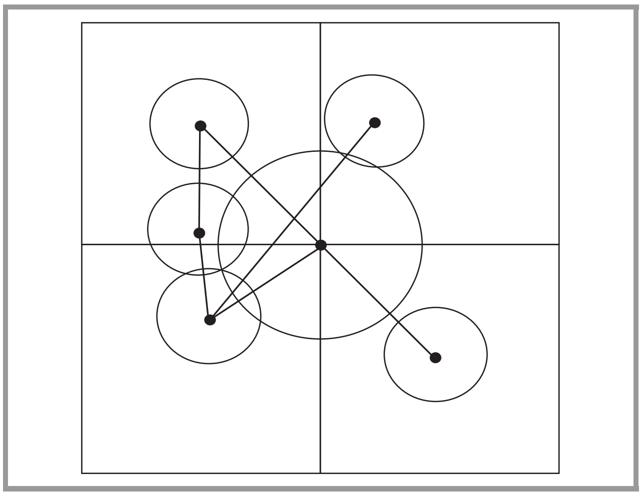

Fig. 10. WSN best layout.

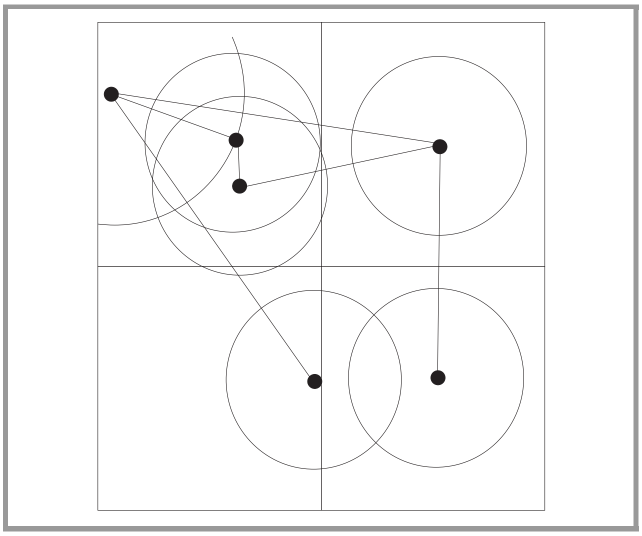

Fig. 11. Application of sensor_range metric to WSN.

coverage inside the overall area that represents the environment in which the WSN operates. Notice that while the

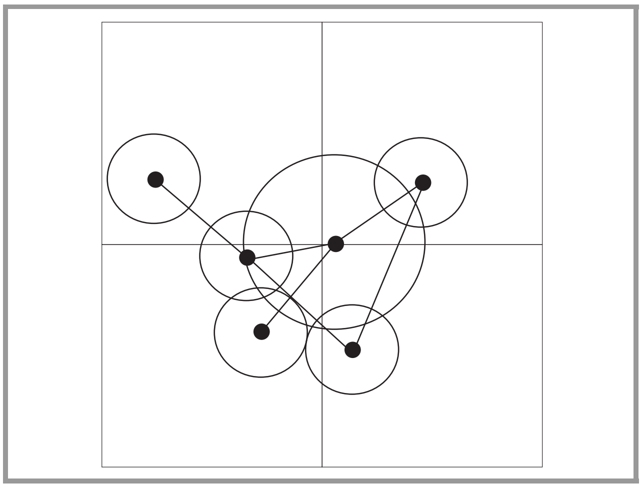

Fig. 12. Application of in_area metric to WSN.

standard metrics automatically still operate on the new formalism, the new ones exploit the new characteristics of the formalism. In consideration of the fact that the length of an arc is in some way a relative measure of the power needed to transmit data between two connected nodes, both these new metrics have been combined with the intersect metric in queries, since arcs crossings generally imply longer paths. The queries are as follows:

```
<?xml version="1.0" encoding="UTF-8"
 standalone="no"?>
<mql rdlref="WSN.rdl"
mdlref="WSN_net.xml">
<result name="sensor_range"/>
<result name="in_area" />
</mql>
```

Results of the application of the new metrics are shown in Fig. 11 and in Fig. 12.

6. Conclusions and Future Works

In this paper we defined a framework and described a tool aimed to solve the problem of graph layout optimization in its most general form by means of Model-driven Engineering, an emerging discipline of software engineering. By means of a formal languages definition every graph based model can be easily inherited from a graph on which layout optimization can be seamlessly performed.

This modeling technique is well defined in the OsMoSys framework, that greatly exalts the modeling features of the approach (by means of OMM) and the solution facilities (by means of OMF). Due to its generality, this approach takes the best from the use of genetic algorithms in order to generate most fit solutions, i.e., best graph layouts.

Several extensions are possible from this work both on framework and tools, extending fitness functions and improving the power of the genetic engine, and on applications, studying other application fields and deepen Wireless Sensor Network example.

References

[1] P. A. Eades, "A heuristic for graph drawing", *Congres. Numeran.*, vol. 42, pp. 149–160, 1984.

[2] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs", *Inform. Proces. Lett.*, vol. 31, pp. 7–15, 1989.

[3] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement", *Software – Pract. and Exper.*, vol. 21, no. 11 , pp. 1129–1164, 1991.

[4] A. Kumar and R. H. Fowler, "A spring modelling algorithm to position nodes of an undirected graph in three dimensions", Tech. rep., Department of Computer Science, University of Texas, 1994.

[5] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing", *ACM Trans. Graph.*, vol. 15, no. 4, pp. 301–331, 1996.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[7] M. K. Coleman and D. Stott Parker, "Aesthetics-based graph layout for human consumption", *Softw. Pract. Exper.*, vol. 26, no. 12, pp. 1415–1438, 1996.

[8] T. Eloranta and E. Makinen, "TimGA – A genetic algorithm for drawing undirected graphs", Tech. rep., Department of Computer Science, University of Tampere, Finland, 1996.

[9] J. Branke, F. Bucher, and H. Schmeck, "A genetic algorithm for drawing undirected graphs", in *Proc. 3rd Nordic Worksh. Genet. Algorith. Appl. 3NWGA*, Helsinki, Finland, 1997, pp. 193–206.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA, USA: Addison-Wesley, 1989.

[11] J. R. Koza, "Survey of genetic algorithms and genetic programming", in *Proc. Western Electron. Show Convent. Conf. WESCON 1995*, San Francisco, CA, USA, 1995, pp. 589–594.

[12] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1996.

[13] S. Kent, "Model driven engineering", in *Proceedings of IFM 2002 – Third International Conference on Integrated Formal Methods*, M. Butler, L. Petre, and K. Sere, Eds. LNCS, vol. 2335. Springer, 2002.

[14] S. J. Mellor, K. Scott, A. Uhl, and D. Weise, *MDA Distilled. Principles of Model Driven Architecture*. Boston, MA, USA: Addison-Wesley, 2004.

[15] J. Bézivin, F. Jouault, and D. Touzet, "Principles, standards and tools for model engineering", in *Proc. 10th Int. Conf. Engin. Complex Comp. Syst. ICECCS 2005*, Shanghai, China, 2005, pp. 28–29.

[16] S. Sendall, R. Hauser, J. Koehler, J. Küster, and M. Wahler, "Understanding model transformation by classification and formalization", in *Proc. 3rd Int. Conf. Gener. Program. Component Engin. GPCE'04*, Vancouver, Canada, 2004.

[17] E. Visser, "A survey of rewriting strategies in program transformation systems", *Electr. Notes in Theor. Comp. Sci.*, vol. 57, pp. 109–143, 2001.

[18] K. Czarnecki and S. Helsen, "Classification of model transformation approaches", in *Proc. 2nd Worksh. Generat. Techniq. Context of Model-Driven Architec. OOPSLA 2003*, Anaheim, CA, USA, 2003, pp. 1–17.

[19] D. Akehurst and S. Kent, "A relational approach to defining transformations in a metamodel", in *Proc. 5th Int. Conf. Unif. Model. Lang. UML'02*, Dresden, Germany, 2002, pp. 243–258.

[20] F. Jouault and I. Kurtev, "On the architectural alignment of ATL and QVT", in *Proc. ACM Symp. Appl. Comput. SAC'06*, Dijon, France, 2006, pp. 1188–1195.

[21] G. Csertan *et al.*, "VIATRA – visual automated transformations for formal verification and validation of uml models", in *Proc. 17th IEEE Int. Conf. Automa. Softw. Engin. ASE 2002*, Edinburgh, Scotland, UK, 2002, pp. 267–270.

[22] A. Agrawal, "Great: A metamodel based model transformation language", in *Proc. 18th IEEE Int. Conf. Autom. Softw. Engin.*, Montreal, Canada, 2003.

[23] I. Kurtev, J. Bezivin, and M. Aksit, "Technological spaces: An initial appraisal", in *Proc. CoopIS, DOA'2002 Federated Conferences, Industrial track*, Irvine, CA, USA, 2002, pp. 1–6.

[24] E. Barbierato, M. Gribaudo, and M. Iacono, Defining Formalisms for Performance Evaluation With SIMTHESys. *Electr. Notes Theor. Comput. Sci.*, vol. 275, pp. 37–51, 2011.

[25] E. Barbierato *et al.*, "Exploiting product form solution techniques in multiformalism modeling", *Electr. Notes Theor. Comput. Sci.*, vol. 296, pp. 61–77, 2012.

[26] M. Iacono, E. Barbierato, and M. Gribaudo, "The SIMTHESys multiformalism modeling framework", *Comp. and Mathem. with Appl.*, vol. 64, no. 12, pp. 3828–3839, 2012.

[27] G. Di Lorenzo *et al.*, "The software architecture of the OsMoSys multisolution framework", in *Proc 2nd Int. Conf. Perform. Eval. Methodol. Tools ValueTools 2007*, Nantes, France, 2007, p. 51.

[28] G. Franceschinis, M. Iacono, N. Mazzocca, and V. Vittorini, "The OsMoSys approach to multi-formalism modeling of systems", *J. Softw. Syst. Model.* , vol. 3, no. 1, pp. 68–81, 2004.

[29] A. Sobeih, W. P. Chen, J. C. Hou, L. C. Kung, N. Li, H. Lim, H. Y. Tyan, and H. Zhang, "J-Sim: A simulation and emulation environment for wireless sensor networks", *IEEE Wirel. Commun.*, vol. 13, no. 4, pp. 104–119, 2006.

[30] M. Li and B. Yang, "A survey on topology issues in wireless sensor network", in *Proc. Int. Conf. Wirel. Netw. ICWN 2006*, Las Vegas, NE, USA, 2006, pp. 503–509.

[31] M. Iacono, S. Marrone, and E. Romano, "Adaptive monitoring of marine disasters with intelligent mobile sensor networks", in *Proc. IEEE Worksh. Environm. Energy, Struct. Monit. Syst.* Taranto, Italy, 2010, pp. 38–45.

[32] M. D'Arienzo, M. Iacono, S. Marrone, and R. Nardone, "Estimation of the energy consumption of mobile sensors in WSN environmental monitoring applications", in *Proc. 27th Int. Conf. Adva. Inform. Netw. Appl. Worksh. WAINA '13*, Barcelona, Spain, 2013, pp. 1588–1593.

[33] M. D'Arienzo, M. Iacono, S. Marrone, and R. Nardone, "Petri net based evaluation of energy consumption in wireless sensor nodes", *J. High Speed Netw.* (to appear).

[34] E. Battista *et al.*, "An integrated lifetime and network quality model of large WSNS", in *Proc. 2nd IEEE Int. Worksh. Measur. Netw.*, Naples, Italy, 2013.

[35] D. B. Jourdan and O. L. de Weck, "Multi-objective genetic algorithm for the automated planning of a wireless sensor network to monitor a critical facility", in *Proc. SPIE Defense Secur. Symp.*, Orlando, FL, USA, 2004, vol. 5403, pp. 565–575.

Mauro Iacono is a tenured Assistant Professor and Senior Researcher in Computing Systems. He received a Laurea in Ingegneria Informatica (M.Sc.) degree cum laude (Hon) in 1999 by Università degli Studi di Napoli "Federico II", Napoli, Italy, and a Dottorato in Ingegneria Elettronica (Ph.D.) degree by Seconda Università degli

Studi di Napoli, Aversa, Italy. He published over 40 peer reviewed scientific papers on international journals, books and conferences and has served as scientific editor, conference scientific committee chairman and member and reviewer for several journals, and is a member of IEEE and other scientific societies. His research activity is mainly centered on the field of performance modeling of complex computer-based systems, with a special attention for multiformalism modeling techniques.
E-mail: mauro.iacono@unina2.it
Dipartimento di Scienze Politiche
Seconda Università degli Studi di Napoli
Viale Ellittico, 31
81100 Caserta, Italy

Stefano Marrone is an Assistant Professor in Computer Engineering at Seconda Università di Napoli, Italy. His interests include the definition of model driven processes for the design and the analysis of transportation control systems, complex communication networks and critical infrastructures.
E-mail: stefano.marrone@unina2.it
Dipartimento di Matematica e Fisica
Seconda Università degli Studi di Napoli
Viale Lincoln, 5
81100 Caserta, Italy