

ISATEM: An Integration of Socioeconomic and Spatial Models for Mineral Resources Exploitation

Fenintsoa Andriamasinoro, Daniel Cassard, and Bruno Martel-Jantin

BRGM - Mineral Resources Division,
BP 36009 Cedex 2, 45060 Orléans, France
{f.andriamasinoro,d.cassard,b.martel-jantin}@brgm.fr

Abstract. In the spatial-integrated socioeconomic model field, a multi-agent approach is appropriate for supporting applications modelled at a detailed territory scale, but it is less used than other approaches when supporting applications modelled at a larger scale. One possibility would be to have a more generic platform that is capable of supporting both levels. Moreover, at the spatial level, integration of dynamic spatial data and, in our case, spatial shape data should be reinforced so that, at any moment during a simulation, a user should be capable of exporting the temporal evolution of the spatial state as a (new) shape map, exploitable in a GIS tool. This paper aims to contribute to the improvement of these two aspects. We began the study with ADK, an existing agent platform, ending up with a new platform known as ISatEM. Our thematic interest is the field of mineral resources exploitation.

Keywords: Spatial shape data, socioeconomic model, simulation, agent, component, mineral resources exploitation.

1 Introduction

Socio-economic modelling can be described as the representation of the relationship between economic activity and social life [10]. Spatial-integrated socioeconomic models (hereafter SISM) are models that often concern the management of natural resources, in which economic or social activities often apply pressure to a geographical space, measured at the territory scale where the social actors are situated. SISM application domains are various: not just water resources [13], bioenergy [16], fisheries [19], etc., but also mineral resources exploitation [1]. Mineral resources exploitation is a SISM composed of socioeconomic actors performing a set of tasks¹, starting from the production of ores, continuing through their transport to the transformation area and ending with the (second) transport of the final product to the area of consumption, where it is either sold and used. The study scale may be micro (local site scale), medium (regional) or macro (national). Socioeconomic impacts are jobs and incomes. Spatial interests are transport distance (impacting on cost), environmental impact (generally due to production and transport) and also spatial (re)distribution of production centres and resources availability [9].

¹ At the application level, some tasks may be abstracted if not necessary for the modelling.

Designing a SISM involves the selection of the appropriate modelling approach, depending on the application territory scale. A multi-agent system (hereafter MAS) [20] is often used, especially if the application territory is defined at a micro scale [5][11]. At a larger scale, other approaches, such as dynamic systems [13], may be more appropriate even if a MAS is still acceptable. Besides, simulating a SISM results in a constant evolution of the state at both socioeconomic and spatial levels. As such, simulation is a way to constantly acquire new temporal data for these levels. In particular, at a spatial level, it should be possible for the simulation user, at any time during an ongoing simulation, to export this dynamic spatial data as (new) shape map data. Consequently, the user, after importing these successive maps in a Geographic Information System (GIS) tool, will be presented with a superposition of files showing the spatial state of one system at different times.

By focusing, at an application level, on the field of mineral resources exploitation, the objective of the work described in this paper is to present and discuss our contribution to two aspects: (a) the improvement of a multi-agent model, formerly designed to support applications modelled at a micro-territory scale, to a more generic object model, capable of also coping with an application modelled at a larger territory scale; and (b) the improvement of the conceptual integration of dynamic spatial shape² data in a SISM, as explained in the previous paragraph. In this paper, shape object refers to these spatial entities having line, circle and polygon forms, and a shape map only refers to ESRI® shape file specification, composed of the three files `shp` (for entity geometry), `dbf` (for entity attributes) and `shx` (for the indexes).

Our approach is to start, as an initial case-study, with an existing multi-agent system known as ADK (for Agent Developer Kit) to finally end up with a new resulting model and simulation platform known as ISatEM (Integrated Systems applied to Exploitation of Mineral resources). Temporal storage of the simulated SISM data is the main issue that must be handled during the work.

Section 2 presents ADK. Section 3 presents our proposal, ISatEM, which is then discussed in Section 0. Section 5 concludes the paper and outlines perspectives.

2 ADK: An Initial Case Study

ADK was developed by Calderoni [5] with the idea of simulating a society of artificial agents. ADK contains three of the main components generally found in Multi-Agent Systems: agents, objects and environment. At a spatial level, ADK uses shape format to model its data. The agent behaviour is based on the triad “perception-deliberation-action”. Messages exchanged inside ADK are then called percepts. At each simulation time step, percepts are propagated throughout the environment according to the previous state of the system. For example, a visual percept, which contains the position of an agent or an object, is propagated so that all agents can behave accordingly, regarding what to do and where to go.

Scope of the ADK approach. The ADK approach is appropriate for applications modelled at a very detailed scale, such as robot foraging problems [5], [11]. In the

² The choice of shape spatial format was made only because it was of interest to us. A discussion about which is better, grid or shape spatial format, is beyond the scope of this paper.

domain of mineral resources exploitation, it is also appropriate for applications such as the exploitation of quartz at the scale of Rantabe in Madagascar [2] or artisanal and small-scale gold mining at the Alga site in Burkina Faso [3] (let *Asgma*). At a larger application scale (cf. Section 3.1), the ADK approach may still be employed, but it may be cumbersome to use the above triad to model an application at such a level: it does not provide more information to the application and it may overburden the simulation performance.

State of the art regarding spatial representation. Regarding the spatial level, the fact that ADK uses shape format is interesting for our purpose but it has some limitations. To ascertain the details of these limitations, let us focus on the application of ADK to *Asgma*.

Asgma is a mineral resources exploitation that is composed of a task sequence, from shaft *extraction* to successive *treatments* of *crushing*, *grinding*, *sluice washing*, *winnowing* and *mill grinding*. Each task is performed by a *team* of actors. Transport is carried out by boys with carts. The *Asgma* model is a SISIM, the socioeconomic part of which is modelled in [3].

The spatial representation of *Asgma* is shown in Fig. 1. In detail, Fig. 1.a shows the content of the unique file, designated *asgma.ads* that ADK uses to initialise the simulation illustrated in Fig. 1.b. Lines starting by # are comments. Only *position*, *nb_agents*, *id* and *dimension* properties concern the spatial initialisation.

The formal syntax of each line in *asgma.ads* is described in Equation 1, in which expressions written in “[]” are optional.

$$\text{classId: (prop}_1 \text{ val}_{11} [\text{val}_{12}]) [(\text{prop}_2 \text{ val}_{21} [\text{val}_{22}])] \quad (1)$$

Fig. 1.b shows agents having a circular shape and task areas having a polygonal shape. We can see that spatial dynamic representation exists, in particular in the transport process: not only can the conveyor agent move but it is also possible to track the spatial evolution of such movement.

However, the spatial information presentation is currently approximate: in *asgma.ads*, the content is set by the user according to his/her own perception. Regarding the conveyor movement in particular, the conveyor goes in a straightforward manner, from the shafts to the crushing area whereas, in real-field, the conveyor cart actually follows determined paths that should allow the model to evaluate the transport cost. These limitations are due to two reasons: not only did real-field data collection in the upstream phase not consider spatial aspects [12], but even if this was the case, ADK cannot read data from a shape map (these two actions should both exist). As a result, the economic impact of the transport activity in the *Asgma* network could not be modelled, as noted in [3].

Besides, ADK displays all polygonal shapes, obligatory in the same form (cf. Fig. 1.b). This is because the *position* parameter only has two values, which actually refer to the barycentre of each task area polygon. All polygon drawings start from this barycentre. All other coordinates that should exist in a polygon are ignored.

Finally, it is not possible for ADK to store the temporal evolution of spatial data during a simulation [3] [5]. When a simulation stops and ADK is closed, all simulated data is lost.

```

#display ----
monitor: (res 10) (per 10) (mdp 30) (out 30000);

#agent environment size ----
field: (dimension 70.0 75.0) (tim 91) (acc 18) (nb_day 9200);

#ore extraction areas (the three shafts on the left) ----
shaft: (position -22.5 20.0) (nb_agents 5) (id shaft1) (type sz) (depth 100);
shaft: (position -25.5 15.0) (nb_agents 5) (id shaft2) (type fm) (depth 75);
shaft: (position -26.5 11.0) (nb_agents 5) (id shaft3) (type fm) (depth 87);

#transport headquarters (in the middle) ----
conv: (position -15.0 22.0) (nb_agents 1) (id conveyor_HQ);

#ore treatment areas (on the right) ----
millg: (position 14.0 8.0) (nb_agents 2) (id mill_grinding_area);
winn: (position 16.0 5.0) (nb_agents 2) (id winnowing_area);
wash: (position 22.0 12.0) (nb_agents 5) (id washing_area);
grind: (position 15.0 17.5) (nb_agents 22) (id grinding_area);
crush: (position 9.0 20.0) (nb_agents 6) (id crushing_area);

```

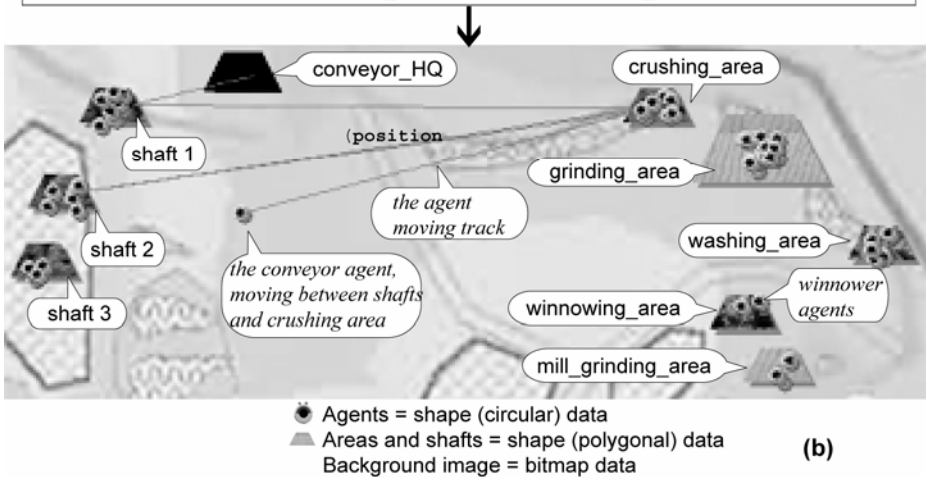


Fig. 1. (a) Content of the file `asgma.ads`, which launches the `Asgma` simulation illustrated in (b)

3 ISatEM: A Proposal

Due to a real-world demand, in the field of mineral resources management, a new application called `FlowMod`, defined at a regional scale, has to be modelled. Since the modelling process at this scale requires going beyond a multi-agent approach, an improvement needs to be made to have a framework that is conceptually capable of supporting both `FlowMod` and `Asgma` applications and of having a better data storage mechanism regarding the previous `ADK` limitations. A collection of real-field data should also be made to avoid an approximate spatial representation, as was explained concerning `Asgma`. Section 3.2 describes how we conceptually translate from `ADK` to

the new framework known as ISatEM (for Integrated System applied to Exploitation of Mineral resources). Beforehand, let us introduce `FlowMod`.

3.1 Presentation of the Larger-Scale Application

Description. `FlowMod` corresponds to the modelling and *prospective* simulation of the production (from a quarry) and the flow management of aggregates³ in Seine-Normandie, a region of France. If we compare `Asgma` and `FlowMod` with regards to territory scale, the whole `Asgma` system is equivalent, in `FlowMod`, to one quarry. Technically, `FlowMod` models the following situation: after aggregates are produced, they are supplied to communities⁴ by trucks. During transport, the shortest itinerary is chosen because the producing companies are obliged to keep transportation distances to a minimum for the large volumes of materials involved in order to remain economically viable. Each quarry has a lifetime after which it is definitely closed, unless the regional administrative authority agrees to re-open it. After a quarry is closed, a new quarry resource and then a new itinerary leading to that resource, has to be searched for.

The dynamic spatial modelling case study presented in this paper concerns the modelling and exportation to a GIS tool of the temporal evolution of the itinerary between a given community and a quarry, knowing that the latter changes over time after a previous quarry is closed and assumed not to be renewed.

Real-field data used. At a spatial level, we acquired the following shape layers, temporarily noted here `layer_name{form}:quarry{point}`, obtained from the ‘GIS for Mineral resources’ project [15], and `community_node{point}`, `road{line}`, and `road_node{point}` layers, obtained from the Route500@ numerical product [18].

At a socioeconomic level, we used the results of the data collection performed in [4], containing, for example, the average income that a quarry exploitation can bring.

3.2 From ADK to ISatEM

ISatEM resumes the essential concepts of ADK (agents, objects, environment, simulation and behaviour), with the difference that the concept of agents and objects is generalized to that of components that may be *spatial* or *transversal*. The ADK environment then becomes an ISatEM environment for components.

The concept of spatial components. A spatial component is a dynamic entity composed of *input ports* (which receive events), *fire ports* (which send events), *behaviour* and *properties*. There are two generic properties: a *spatial property* designated `Geometry`, which may be a *line*, *circle* or *polygon* and a *behavioural property* designated `state`. The possible values for `state` are contained in a static property designated `possible states`. All non-generic properties are either *socioeconomic properties* or *component properties*, i.e. properties referring to other components. `Geometry`, `state`

³ Aggregates are unconsolidated materials used as one of the main components of concrete and as foundations for road and railway infrastructures.

⁴ In France, a “community” corresponds to an administrative unit at a town scale.

and *socioeconomic* properties are called *temporal properties*, namely properties whose value can evolve over the simulation time.

An output fire port takes the form of `FireEvent()`, where `Eventx` is the event to be sent from that port. Symmetrically, an input event port is associated with the function that should be performed on receiving the associated event. It takes the form of `On-EventX()`. There is *always* at least one input port named `OnTimeChanged()`, which continuously receives the system timer event at each simulation time step. The current value of `state` determines the *unit* goal and then the unit behaviour (which may include a thinking process if the component represents a cognitive agent) to be adopted during that time step, after which `state` is re-evaluated by a `StateTransition()` function, invoked after `OnTimeChanged()`.

In `FlowMod`, spatial components are `Quarry` (circle), `RoadNode` (point), `Road` (line), `Community` (circle), `Itinerary` (line) and `Truck` (polygon). It may be noted that only the four first components have a corresponding shape map of real-field data.

Fig. 2 illustrates what a spatial component is. As an example in `FlowMod`, the `Quarry` component is shown (in a simplified way, for space reasons).

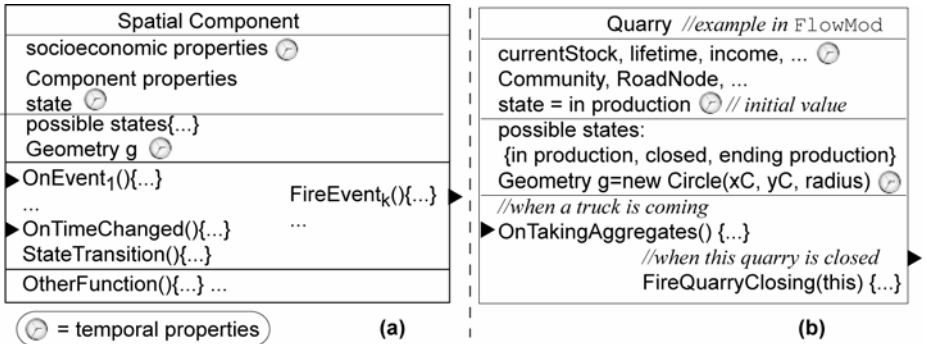


Fig. 2. The generic scheme of a spatial component (a), followed by an example from the `FlowMod` application: the `quarry` component (b)

From the behavioural points of view, a `quarry` is an entity in which behaviour is driven by the goal to produce the necessary aggregates needed by its corresponding community while avoiding having too much stock of non-consumed production. Likewise, a `truck` conveys the maximum of aggregates in a day to maximize its profits while trying to diminish as much as possible the quantity of CO₂ produced, and a `community` tries to acquire the appropriate quantity of aggregates corresponding to its population’s needs (local building of houses or roads, etc).

Transversal components. A transversal component is a component with the following differences compared to a spatial component: (a) it has a global view to all spatial components, allowing it to perform a computation involving all of the latter elements, and (b) it does not have `state` and `Geometry` properties.

In `FlowMod`, transversal components are:

- `ItinerarySearch`, which looks for the itinerary and the quarry that should supply a community requiring aggregates: research criteria are the existence of a non-used quarry and the shortest itinerary between it and the community.
- `TruckManager`, which adds or removes a truck, depending on the quantity produced for a community.
- `CO2Evaluator` (and resp. `FlowEvaluator`), which computes, at each `OnTimeChanged()`, all the CO₂ released by the production and the transport processes (resp. that computes the quantity of aggregates conveyed by trucks).

During a simulation, the `OnTimeChanged()` of all transversal components is always run before that of spatial components.

3.3 A Global View of the ISatEM Architecture

At a generic level, ISatEM is composed of three packages: a *kernel*, a *user interface* and a *document*, as detailed below and illustrated in Fig. 4.

The kernel. The ISatEM kernel is based on three modules: a *library module*, a *model module* and a *simulation module*. One module corresponds to one stage of an application design, and is inspired by the architecture proposed by [17].

The *library module* is like a factory that creates all the types of components required by the application. It only is an environment development where the code of the components is written and compiled. We will not go further into this module here.

Model module instantiates the component types designed in the previous module. There are three ways to instantiate a component: either manually, or by computation during a simulation, or by a (shape file) importation process, via an *importation/exportation engine* (let \mathbb{I}/\mathbb{E} engine) included in this module. Created components may be in interaction with each other or not, depending on the application.

Simulation module simulates the content of the *component environment*. There are three modes of simulation: the *none mode*, during which nothing is saved during the simulation, the *execution mode* during which, at each time step of the simulation, the value of each temporal property of each component is saved “online” by a *property logger*, and the *replay mode*, which reviews the simulation at the “offline” phase, only by retrieving the corresponding property values of the current simulation time, previously saved. In that mode, it is possible to go back and forth during the simulation, without performing any computation from the beginning again.

The user interface (UI). This contains the interface of the functions described above. Furthermore, it contains (a) a *map displayer*, which shows all shape maps imported by the \mathbb{I}/\mathbb{E} engine via a *layer manager*, and (b) a *comment editor*, at the user’s disposal. Simulation and model modules have the same UI as is presented in Fig. 3.

The ISatEM document. An ISatEM document contains all the data corresponding to an application. When the ISatEM software starts, no application is actually loaded until the ISatEM document containing that application is opened by the user, (exactly like a user opens an MS-Word® document in MS-Word® software). An ISatEM document is a container, in the form of a ZIP file, generated by ISatEM, and containing the following files:

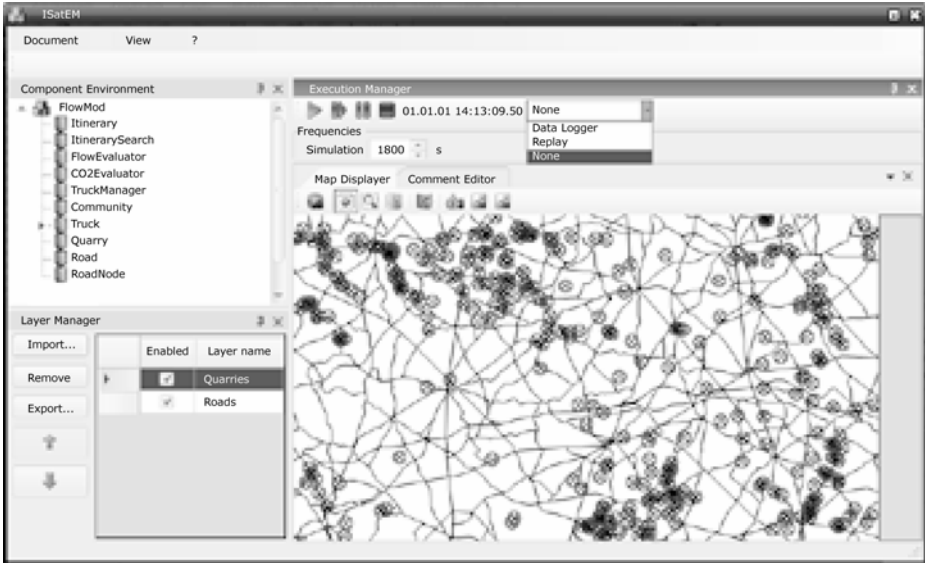


Fig. 3. The user interface of both model and simulation modules. Are displayed: at top left, all FlowMod components; at bottom, two shape layers (quarries and roads); at top right, the simulation time as well as the simulation time step and the three simulation modes.

- The files generated/used by the *property logger* during the execution mode. There is one file per component, as formalized later in Equation 2. The files are in CSV (Common-Separated Values) format.
- A list of the current imported shape files.
- The files currently managed by the comment editor. They are in HTML format.
- A file storing the current order in which layers of imported shapefile are successively displayed, from top to bottom, as in a GIS tool. For example, in Fig. 3, the quarry layer is above the road layer (cf. the layer manager view).
- All classes designed by the library module. They are stored in a simple object (binary) file, to which we allot the extension .LBR (referring to the *library* term).

Organisation of the property logger files. The organisation of the information in the CSV file managed by the property logger is inspired from that of *asgma.ads* with, in addition, the introduction of temporal values. However, one CSV file corresponds not to the whole simulation, as in *Asgma*, but to one component only.

Equation 2 explains the process, where expressions in “[]” are optional. As an example: for a quarry (a circle shape) designated *q2*, having an initial stock of 1000 tonnes, and being capable of producing 0.5 tonnes of aggregates per simulation time unit, we will have the information below inside the file named *quarry.q2.csv* after one simulation time step, and under the execution mode.

```
0, stock, 1000
0, circle, 700, 240, 4
1, stock, 1000.5
1, circle, 700, 240, 4 // the same because the quarry does not move
```

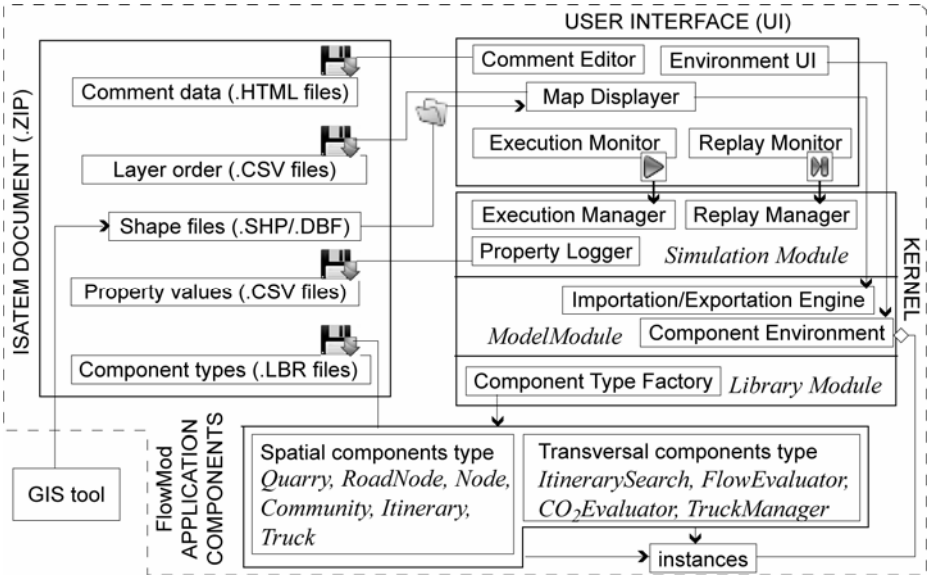



Fig. 4. Illustration of the general architecture of ISatEM (the area inside the dashed line)

```
// Recall of Equation 1, applied to asgma.ads
classId: (prop1 val11 [val12]) [(prop2 val21 [val22])]

// Transformation for the CSV file
classId
    → a CSV file named classId.componentId.

(prop1, val11, [val12])
    → currentTime, prop1, val11, [val12]
    → and written on a whole line

// Regarding the Geometry property, we have
currentTime, "circle", xCentre, yCentre, radius // for circles
currentTime, "polygon", x0, y0, x1, y1, x2, y2 [, ..., xn, yn] // for polygons
currentTime, "line", xa, ya, xb, yb [, ..., xm, ym] // for lines
```

3.4 Generation of New Dynamic Shape Data

The only mechanism that drives the dynamic (i.e. the evolution of the system state) of ISatEM is the simulation of the content of the component environment. The full process is presented in Fig. 5 and explained in the next paragraph.

If the user wants to evolve the state of entities contained in a shp/dbf file, he/she has to (a) import the map into ISatEM via the I/E engine and (b) select the area of interest (let *a_{oi}*) from the maps, via the *map displayer*. This selection transfers the content of each entity in the shp/dbf files to a created instance of the corresponding spatial component as follows: the shp content is transferred to the *Geometry* property

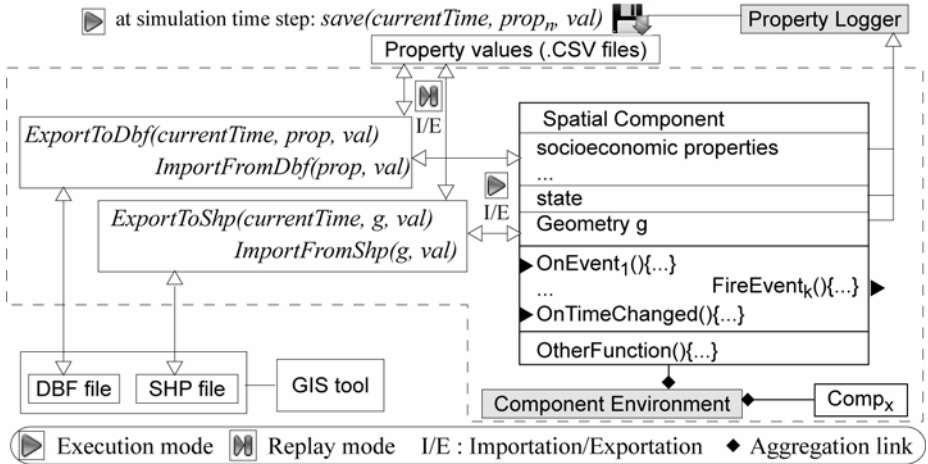


Fig. 5. The mechanism of the importation/exportation engine (the area inside the dashed line)

of that component and the `dbf` content to the socioeconomic properties. For example, the content of a quarry entity from the `shp/dbf` layer is transferred to an instance of `quarry` component, a sample of which has been described in Fig. 2.a. Next, during simulation, values of these spatial components will evolve via their interaction to other spatial components and via the action of transversal components. At any time, the temporal property values of these spatial components can be exported to a new shape file.

As a result, a user can export two types of (new) shape data layers, classified in the way the corresponding upstream components have been created:

- *Existing evolved layers*, viewed as the temporal evolution of the initial imported shape, at the time t where it is exported.
- *New layers*, resulting from the exportation of instances of spatial component types that are not initialised by a previous importation process but by computation during the simulation. It is, in `FlowMod`, the case for the `Itinerary` (cf. Fig. 6, simplified for space reasons) and `Truck` layers.

Fig. 7 illustrates the general shape data generation mechanism, from the importation of layers to the generation and exportation of the new successive `Itinerary` layers over time. The left part of Fig. 7 is an abstract view of Fig. 6.

Note that the `state` property is not manipulated by the I/E engine. `state` is pre-initialised in the component type declaration (e.g. in the `quarry` component, the initial value of `state` is `in_production`, as illustrated in Fig. 2.b), is saved by the property logger during simulation, and is not concerned by the exportation process.

3.5 Results

Let us take a scenario where, each time an aggregate supplying the community of Rouen (Seine-Maritime) has produced over 100 days it is closed and is not renewed.

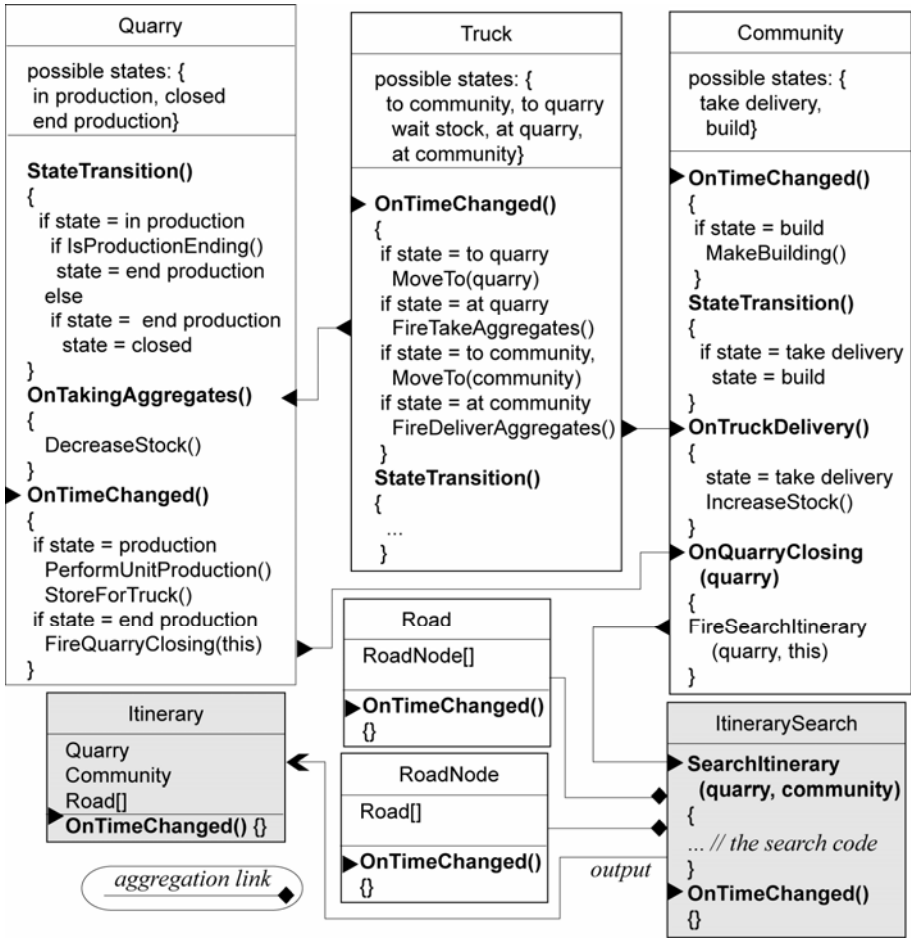


Fig. 6. Block diagram of how an Itinerary component is generated during simulation

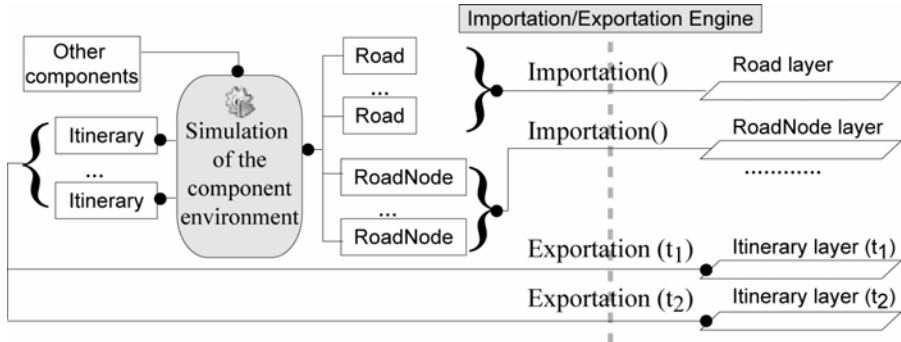


Fig. 7. Mechanism for generation of the Itinerary shape layers over time, starting from the importation/exportation engine, to the simulation of the content of component environment

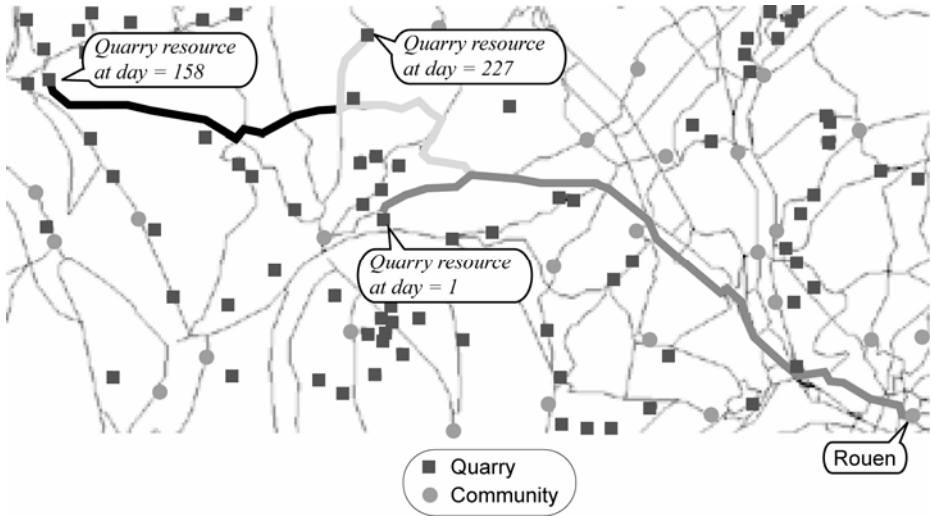


Fig. 8. Result of the exportation of three itinerary layers to three successive resources, obtained during one simulation. The start of the itinerary is the Rouen community.

As explained before, this situation systematically involves the search for a new quarry and itinerary after 100 days. At each newly found itinerary, the *Itinerary* layer is exported and integrated in a GIS tool where, for better visibility, only the itinerary starting from Rouen is kept. In ISatEM, time for exportation is not important as long as exportation is made inside the itinerary lifetime. The simulation time step is 30 minutes. Fig. 8 shows the final result visible in the GIS tool, after three exportations.

4 General Discussion

The result illustrated in Fig. 8 is still being adjusted and validated by the domain experts, regarding the itinerary search rules (the quarries found are not always exactly those expected by them) and the simulation lifetime: at present, we are able to stabilise tests of up to around 15,000 time steps (i.e. 300 days of 30 minutes) with the replay mode. However, this current validation does not affect the capacity of the system to generate new shape data by simulation, whatever the simulation mode.

In a more general manner, Isatem can be viewed as a complement to a GIS tool. It generates new dynamic spatial information by simulation, can display such information by itself (e.g. truck moving) and can store and reproduce the action (via the replay mode). However, Isatem cannot analyze spatial data to the same extent as a GIS. Inversely, a GIS can represent spatial information but in a static way only. One solution is to make them operate in close collaboration since, in the end, they are both data providers and analyzers at their respective levels. As one result, the quality of a map to be produced as a decision aid support, for example, can be the result of its iterative design between both tools.

Returning to the modelling work, and compared to the ADK state of the art as described in Section 2, it is now possible to import and export shape files. Besides, spatial and socioeconomic temporal data can be saved and restored at any time. At a methodological level, this improvement has been achieved thanks to the improvements made from (a) the `asgma.ads` file syntax formalized in Equation 1, to (b) the `quarry.q2.csv` file (taken as example) formalized in Equation 2, and to (c) finally the concept of the ISatEM document. Nevertheless, we agree that choosing `.CSV` as the format to store the components' property value may be open to discussion. In fact, our first idea was to apply the document concept. A better optimized format for saving temporal values undoubtedly exists and should be a subject for further research. Besides, if ISatEM appears overall to be conceptually better than ADK, one feature that ADK currently only has compared to ISatEM is, at a display level, the ADK capacity to follow the track of a moving entity (cf. Fig. 1). Integrating this feature into ISatEM needs to be investigated. Integrating a generic track layer may be a first manner of approaching this situation.

Regarding SISM design, generalizing the concept of agents and objects of ADK to ISatEM components has the following advantage⁵: two application territory scales can be managed. Indeed, if `FlowMod` can be modelled with ISatEM components, using the ISatEM component to model ADK architecture (recall the presentation in Section 2), then the `Asgma` application is still feasible. The result is shown in Fig. 9 where a `PerceptsPropagator` is transversal and an `ADKAgent` is spatial.

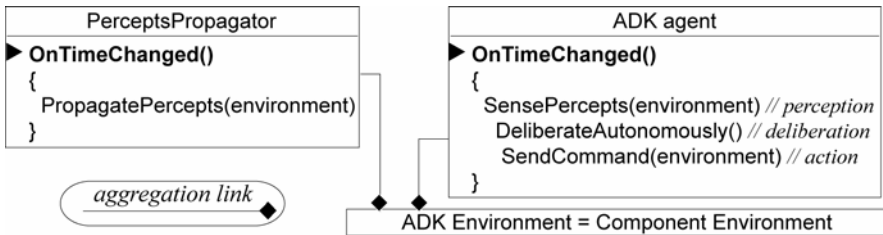


Fig. 9. Design of the ADK architecture using the ISatEM component. The ‘perception-deliberation-action’ triad as well as the propagator concepts of ADK are still present.

The notion of components may lead the reader to think about the DEVS (Discrete Event System specification) components [8], widely used in modelling and simulation contexts, integrating space [e.g. 19]. The reason we do not use DEVS is a visualization issue: we want to see the *continuous* movement of our trucks in the system, as has been done with the conveyor in `Asgma` (cf. Fig. 1). We think that DEVS is not the appropriate approach for that purpose, unless the duration of the internal state transition of the DEVS component is statically set to that of the simulation time. But in such case, the actual interest of using DEVS formalism is diminished.

⁵ Another interest of this change is that it offers the possibility of supporting other SISM applications based on approaches other than a multi-agent approach, such as for example the system dynamics approach as used by [13].

5 Conclusions and Perspectives

Summary. Spatial-integrated socioeconomic models (SISM) are models that can represent the relationship between economic activity, social life and the pressure social actors apply to the geographical space containing them. Although a multi-agent approach is often used to support applications modelled at a detailed territory scale, it may be less appropriate when an application is modelled at a larger scale. Moreover, since SISM is a spatial model, conceptual integration of dynamic spatial data and, in our case, spatial shape data should be reinforced.

By focusing on the domain of mineral resources exploitation, we have proposed, in this paper (a) to improve a multi-agent model to obtain a more generic object model that is capable of also supporting applications modelled at a larger scale, and (b) to integrate the mechanism of dynamic spatial shape data, in which the temporal evolution of the spatial state of an ongoing simulation can be exported at any time, as (new) shape data stored in a map exploitable in a GIS tool. Our approach was to start from an existing multi-agent platform known as ADK, ending up with a new platform known as ISatEM.

Perspectives. Although ISatEM seems to be a better result for us, regarding dynamic spatial modelling, ADK still has the capacity to follow the track of a moving component, a feature that ISatEM does not yet integrate. The integration of such a feature constitutes the first perspective for ISatEM. The other perspective is the integration of what we call *indicator display*. An indicator is one component property, the evolution of which a user wants to visualize via one feature of the shape representing this component in a screen. A shape feature may be a *width* (for a line), a *surface size* (for a polygon and circle), or a *colour* (for all of them). Features may be combined if possible. During a simulation, when an indicator evolves, the corresponding shape feature also evolves. It results in a display of dynamic spatial shape data information, even on non-moving shapes. For example, the colour of a quarry shape may evolve according to its current stock (the indicator). Regarding the mineral resources field, one technique of selecting indicators is presented in [6].

References

1. Andriamasinoro, F., Angel, J.M.: High-Level vs. Low-Level Abstraction Approach for the Model and Simulation Design of Ultra-pure Quartz Exploitation in Northeastern Madagascar. In: Joint Conference on Multi-Agent Modelling for Environmental Management (CABM-HEMA-SMAGET), Bourg St-Maurice, France, March 25-28 (2005)
2. Andriamasinoro, F., Angel, J.M.: Modeling the Ultra-pure Quartz Exploitation in Northeastern Madagascar: Impact of the Activity on the Socio-Economical Situation of the Population. *Journal of Socio-Economics* 36(2), 311–329 (2007)
3. Andriamasinoro, F., Jaques, E., Pelon, R., Martel-Jantin, B.: Artisanal and Small-scale Gold Mining in Alga (Burkina Faso): Building a Decision-Aid Model for Development and Governance. In: The Summer Computer Simulation Conference (SCSC 2005), Philadelphia, Pennsylvania, USA, July 24-28, pp. 292–297. CD-Rom (2005), ISBN 1-56555-299-7
4. Andriamasinoro, F., Orru, J.F.: Collecte de données socioéconomiques sur les exploitations de granulats dans la boucle d'Anneville-Ambourville (Seine-Maritime). Technical report, BRGM/RP-57465-FR, 105 pages (2009)

5. Calderoni, S.: Ethologie Artificielle et Contrôle Auto-Adaptatif dans les Systèmes d'Agents Réactifs: de la Modélisation à la Simulation. PhD Thesis, University of La Réunion, 175 pages (2002)
6. Chamaret, A., Récoché, G., O'Connor, M.: Proposal for a top-down/bottom-up approach to build up indicators of sustainable development for use in the mining industry in Africa. In: The SDIMI conference 2005, Aachen, May 18-20 (2005)
7. Courdier, R., Guerrin, F., Andriamasinoro, F., Paillat, J.M.: Agent-based simulation of complex systems: application to collective management of animal wastes. *Journal of Artificial Societies and Social Simulation* 5(3) (June 30, 2002)
8. Dalle, O., Zeigler, B.P., Wainer, G.A.: Extending DEVS to support multiple occurrences in component-based simulation. In: 40th Conference on Winter Simulation, Miami, Florida vol. 933-941 (2008), ISBN:978-1-4244-2708-6
9. Deschamps, Y., Milesi, J.P., Hocquard, C., Bouchot, V., Salpeteur, I., Pelon, R., Ralay, F.: Germanium potential in Africa: a predictive approach methodology. In: 21st CAG21 - Colloquium of African Geology, Maputo - Mozambique, July 03-05 (2006)
10. Haberl, H., Gaube, V., Díaz-Delgado, R., Krauze, K., Neuner, A., Peterseil, J., Plutzer, C., Singh, S.J., Vadineanu, A.: Towards an integrated model of socioeconomic biodiversity drivers, pressures and impacts. A feasibility study based on three European long-term socio-ecological research platforms. *Ecological Economics* 68(6), 1797-1812 (2009)
11. Hugues, L., Drogoul, A.: Synthesis of Robot's Behaviors from few Examples. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2002 (2002)
12. Jaques, E., Zida, B., Billa, M., Greffié, C., Thomassin, J.F.: Artisanal and small-scale gold mines in Burkina Faso: today and tomorrow. In: Hilson, G.M. (ed.) *Small-Scale Mining, Rural Subsistence and Poverty in West Africa*, ch. 10, pp. 114-134 (2006)
13. Lanini, S., Courtois, N., Giraud, F., Petit, V., Rinaudo, J.D.: Socio-hydrosystem modelling for integrated water-resources management – The Hérault catchment case study, southern France. *Environmental Modelling and Software* 19(11), 1011-1019 (2004)
14. Le Page, C., Bousquet, F., Bakam, I., Bah, A., Baron, C.: CORMAS: A multiagent simulation toolkit to model natural and social dynamics at multiple scales. The ecology of scales, Wageningen, the Netherlands, June 27-30 (2000)
15. Leistel, J.M.: SIG Ressources Minérales Françaises - Architecture et mode d'emploi des applications de saisie. Technical report, BRGM/RP-54545-FR, p. 67 (2005)
16. Madlenera, R., Domach, J.: Introduction to the special section Modeling socio-economic aspects of bioenergy use: methodologies, case studies, and policy evaluation and guidance. *Energy Policy* 35(12), e1-e4 (2007)
17. Serment, J., Espinasse, B., Tranvouez, E.: For a generic software architecture facilitating environmental decision support system development: Illustration with the Camargue Ecosystem. In: The Joint Conference on Multi-Agent Modelling for Environmental Management (CABM-HEMA-SMAGET), Bourg St-Maurice, France, March 25-28 (2005)
18. The Route 500® numerical product,
http://www.cartosphere.com/en/ign_route500.htm
19. Versmisse, D., Macher, C., Ramat, E., Soulié, J.C., Thebaud, O.: Developing a bio-economic simulation tool of fisheries dynamics: a case study. In: Proceedings of the International Congress on Modelling and Simulation, International Society for Computer Simulation, Christchurch, New Zealand, December 10-13 (2007)
20. Wooldridge, M.: An Introduction To Multiagent Systems, p. 340. John Wiley & Sons, Chichester (2002)