MASC: Map Sectors Creator
A Tool to Help at the Configuration of Multi-Agents Systems for Everyone

Aurélie Gaudieux[1], Joël Kwan[2], Yassine Gangat[3] and Rémy Courdier[3]
[1]*Centre d'Economie et de Management de l'Océan Indien, University of Reunion Island, Saint-Denis, Reunion Island*
[2]*Ecole Supérieure Ingénieur Réunion Océan Indien, University of Reunion Island, Saint-Denis, Reunion Island*
[3]*Laboratoire d'Informatique et de Mathématiques, University of Reunion Island, Saint-Denis, Reunion Island*

Keywords: Automation, Initialization, Configuration, MAP, Sketches, Plans, Cutting Out, Extraction, Grid, Bitmap File, Code Snippet, Online, Web Based, User-Friendly.

Abstract: The initialization at the beginning of all type of simulation is recurring. In the field of multi-agents, spatial environments on a micro or macro scale, or more abstract context, such as mapping out energy sources for example, need often to be modelled. The initialization of these environments wherein evolve agents is generally tedious and time-consuming depending on the fineness of the mesh of the cutting in case of a 2D spatial representation. In this paper, we present MASC (Map Sectors Creator), a user-oriented tool that permits easy initialization through the creation of a mesh directly usable in multi-agents systems simulations platforms. The automation of cutting out maps followed by the generation of directly usable code snippets optimize the working time on initialization and allow to focus on the simulations results and observations.

1 INTRODUCTION

Computer modelling is the formal description of a system that allows manipulation of a virtual copy using a computer. To obtain this virtual representation, a model must be implemented. The process from a theoretical context to a more concrete one involves the production of codes to be used to have a final product in the form of an application.

Before running a simulation, the initialization must be settled, which includes configuring the simulation context. Knowing that the context has a huge impact on the proceedings and the outcomes of the simulation, we often give more attention to data in particular and starting environment. Each configuration leads, so to speak, to an almost unique experience depending on the configuration settings (Gangat, et al., 2009).

Optimization of working time issue arises when handling complex systems, especially in multi-agents systems where more attention must be paid to data and different proceedings performed during iterations.

To help solving this problem, we propose MASC – Map Sectors Creator, an easy to use tool that can provide an environment base represented by cutting a base map, a diagram or a plan into a grid of sectors directly usable in multi-agents simulation platforms such as NetLogo (Wilensky, 1999) or the platform of multi-agent simulation of IREMIA laboratory at the University of Reunion Island, GEAMAS-NG (GEneric Architecture for MultiAgent Simulations) (Payet, et al., 2006).

Thus, MASC permits to save time and focus on implementation of models. In addition, MASC provides a simplified user experience through opportunities brought by recent developments in Web technologies. We want to propose a user-friendly tool that could help anyone to optimize working time and facilitate the configuration of the initialization. The word "masc" means "men" in Portuguese.

This kind of tool is useful for several simulation domains for example geographical, economic, meteorological or epidemic ones.

2 MASC

2.1 The Need

In the context of models based on real existing ones, we perform simulations in which agents operate on a

Gaudieux A., Kwan J., Gangat Y. and Courdier R..
MASC: Map Sectors Creator - A Tool to Help at the Configuration of Multi-Agents Systems for Everyone.
DOI: 10.5220/0005133408360844
In *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (MSCCEC-2014), pages 836-844
ISBN: 978-989-758-038-3

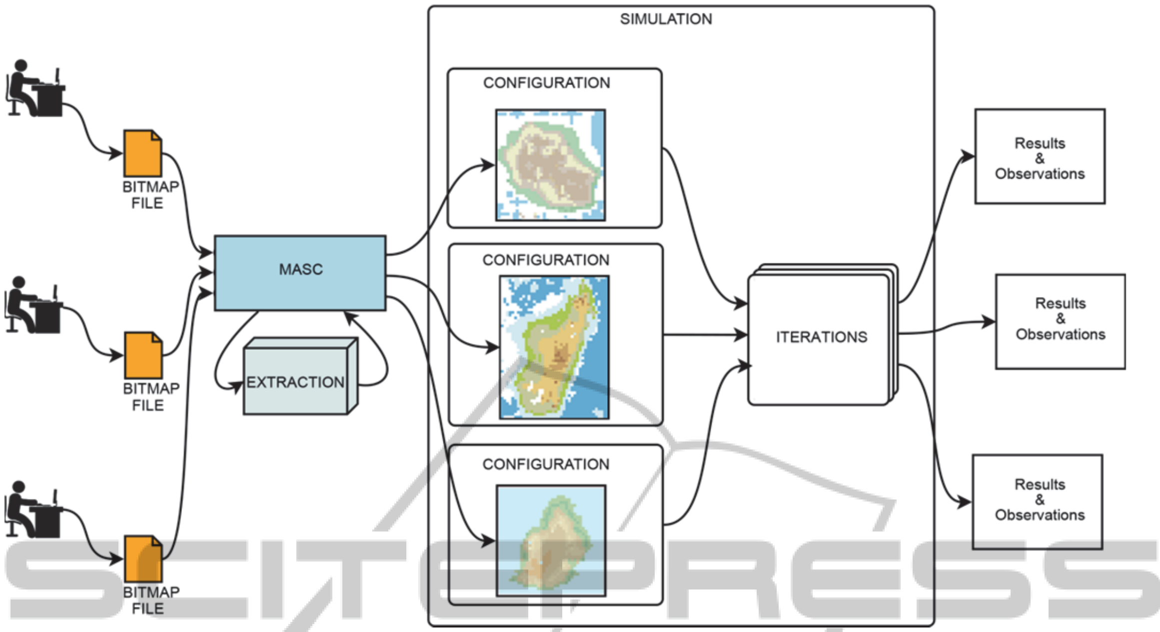

Figure 1: Using MASC for configuring multiple experimentations of a simulation.

space generated from spatial division of maps. E.g.: SIEGMAS (Gaudieux, et al., 2014).

Some existing tools already provide some parts of solution to this problem. We can take the example of XELOC Language, eXtensible Editing Language Of Configuration, which is a support for the configuration an initialization of multi-agents systems. XELOC is based on the basics of XML and requires the handlings of another programing language. One of MASC objective is to repeal both the acquisition processes and handling of a new tool.

The development of a tool like MASC was driven by mainly two reasons.

Firstly, the need of a tool allowing the automation of the process of cutting out maps to perform multiple simulations of a same model with different geographical areas. The tool aims to facilitate the transition from map into an exploitable representation. It is very interesting to incorporate modelling with different configurations. Some models can lead to significant results from the abstract level to a less abstract one.

And secondly, the use of a user-friendly tool also permits to effortlessly make rather redundant tasks of creating an environment where agent evolves along different simulations. Several configurations can then be produced more efficiently in no time as revealed in Figure 1.

2.2 The Concept

MASC has been designed to increase productivity on implementing a model so that it provides a streamlined user experience. That is why the tool was developed in order to ask the fewest operations and minimum familiarization while providing enough controls. The user interactions with MASC interface are then based on the daily operations being done by any computer user (drag and drop, file uploading, form filling, mouse clicking). The graphical user interface (Figure 2) is quite simple and reduces the displayed information to only the useful ones.

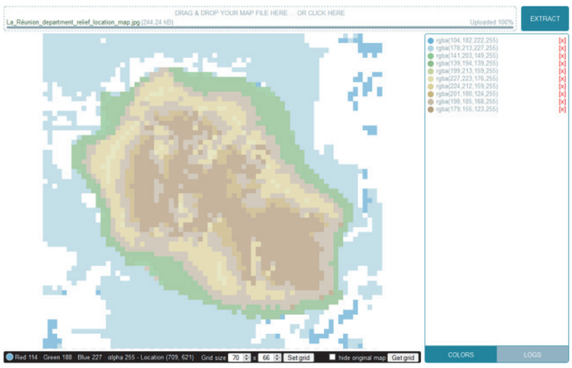

Figure 2: MASC GUI.

The quantity of requested information to input is also kept to a minimum, the rest being automatically filled by calculations.

A question arose as to which file types would be supported by the tools. The most used type of output images are vector graphic files generated from GIS softwares or bitmap files for several maps, plans, drawings, sketches, or other digital copies created from paper versions. Knowing that it is much less restrictive to convert a vector image into bitmap on to the opposite direction, we have decided that the first versions of the tool would only support bitmap images in input.

Moreover, we wanted the tool be designed so that it is usable at any time by anyone with any computer without altering the workspace of the user. That is why we chose to make an online web based application.

Regardless the necessity of a connection to the server, the downside of this type of application is that the power of calculation is limited by the capacity of the hosting server and the technology used for the different calculations and the allowed uploading file size. For the first version of our application, we chose to make it deployable to a maximum of easily accessible servers. So we deployed it on a basic server provided by a common hosting plan. And for a maximum compatibility with available programming languages on these type of hosting, we used PHP programming language for the server side coupled with HTML5 and JavaScript for the client side.

On the upside, anyone can access the application with a simple connection to the server and a recent Internet browser. In a future perspective, the server side will be switched to a Java version that will have a better performance calculations but require an upgrade of the hosting plan.

2.3 Presentation

2.3.1 The Ease of using MASC

Using MASC is divided around three main steps. The first step consists of inputting all the necessary information needed by the tool to correctly extract a cutting grid space from an image that is presented to it. This stage takes place mainly on the client side that is to say on the user's Internet browser (shown highlighted in Figure 3). The information required for the proper functioning of the tool are:

- A bitmap image file in JPEG, PNG, GIF, or TIFF format, which is a representation of the space where agents evolve during the simulations.
- The size of the grid is simply summarized by the number of sectors to cut out across the width of the image. Because we are working with a grid where sectors are square shaped, the number of sectors in the height direction can be automatically calculated.
- A list of colors selected from the preview of the image taken by clicking on it.

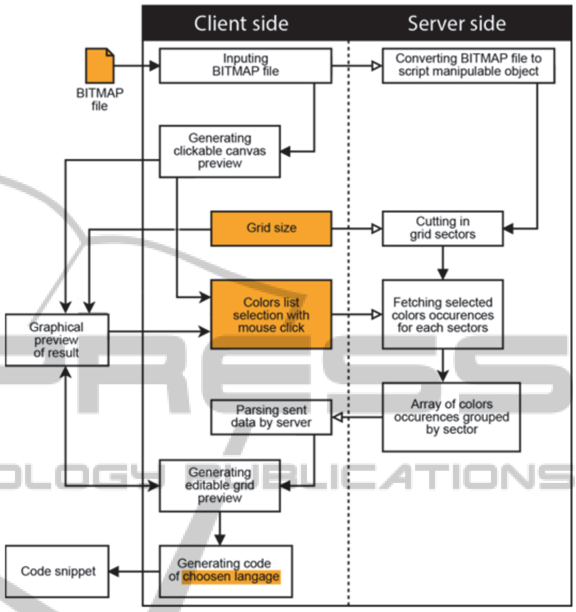

Figure 3: Details of MASC workflow.

In a second time, these three data are sent to the server. Then the script on the server can manipulate the bitmap file and can extract a grid that will serve as a starting point for generating the code snippet used for initialization of simulations. At this stage, the user will have no operations to do, all the treatments will be done in autonomy at the server level.

In the last step, new information resulting from the extraction made on the server side is displayed graphically to the user using the preview area of the graphical user interface displayed on his browser. On the one hand, the user has a preview and on the other hand he can slightly edit the grid before final code snippet generation according to the chosen programming language. At this level, the user can still change the color of the different sectors by clicking on them.

If needed, the user can change the number of given sectors across the width to change crisscross scaling and then regenerate the code snippet.

2.3.2 Colors

Always with the idea of putting the user at the center of the experience and increasing the productivity in mind, MASC uses a list of colors selected from the image by the user. This list is used in the second step. It also allows to filter unwanted colors and to avoid to keep annotations that are often merged on the image itself. So the user only select interesting colors.

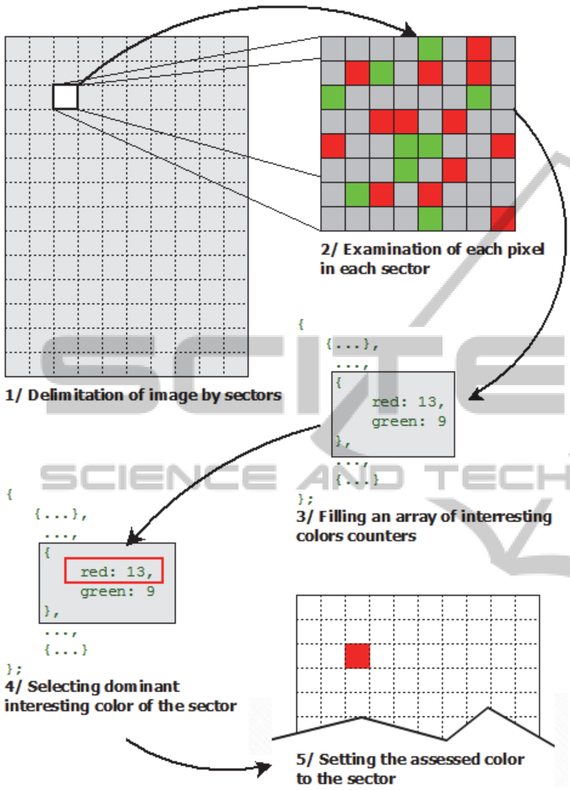

Figure 4: Colors interpolation.

When extracting the colors, the image is divided into sectors according to the given cutting grid. Then each pixel is examined to determine if the pixel color matches with one of the colors in the list. If it is the case, the counter of the matching color in the sector of the pixel is incremented.

In this process, the tool tolerates a degree of shade. Thus, the color alterations brought by the image compression does not prevent MASC to correctly extract the desired colors in each area.

An array of colors grouped by sector is gradually created regarding the progress of the review of the image pixels. Then, the dominant color is selected to be the color of the sector (Figure 4). By proceeding pixel by pixel, the precision of the interpolation that assesses a sector color is dependent on the cutting out grid size. The more sectors across the width is given in input, the greater the precision will be. There is no minimum grid resolution, it all depends on the goal and the need of the model.

To allow a certain degree of freedom and to leave the choice of colors to prioritize on sectors, the user can click on the preview grid cells to choose the dominant color regarding the listed ones in the working sector. These modifications are stored in the client side to avoid the useless data transfers between client and server. So directly after the adjustments brought by the user, the code snippet containing the representation of the space can be generated.

3 CASES STUDY

The team being at work on two ongoing projects. SIEGMAS which use the simulation platform NetLogo and EDMMAS, which use the GEAMAS-NG, we would like to test the relevance of MASC in these projects. In this part of the paper, we will explain the advantage of using MASC with theses example.

3.1 SIEGMAS

SIEGMAS (Stakeholders Interactions in Environmental Governance by a Multi-Agent System) is a modelling conceptualized in order to study the deviances of the farmers against the forester and natural resources protected by the management's transfers or another text of law.

3.1.1 Context

In the context of this project, the economical and classic steps being limited enough for the conceptualization of the individuals' behaviors and the process of the common pool resources, a process of multi-agents systems allows some improvements in the study of these phenomenon and the creation of a sales tool for the decision

This modelling has been conceptualized in order to study the deviances of the farmers against the forester and natural resources protected by the management's transfers or another text of law.

Therefore, this simulation allows to identify the interactions between the actors of the governance taking part in the management's transfers and to propose some targeted measures in agreement being able to reduce the merging anomalies. Thus, this model reveals that the knowledge of the sanctions by farmers does not prevent them to infringe the legislative measures of protection of natural resources coming from the management's transfers or located in some few protected areas. (Brandouy, et al., 2012).

3.1.2 SIEGMAS

Farmers (deviant or not) generally aim to make their land as profitable as possible to cover several costs (e.g. investment, maintenance). A farmer can choose to spread legally his cultivation via the purchase of new plots or reaching the legally maximum allowed use of natural resources. In fact, the management's transfers grant the farmer through the basis community some right of cultivation of some natural resources. However, the farmer can be a deviant by farming illegally some plots of natural and forester resources.

This model is mainly focused on the farmers and the government and their interactions in relation to the forester and natural resources as well as the resources produced on the farms.

The model SIEGMAS has been implemented under the platform of the programming called NetLogo. The farmers can respect or infringe the legislation towards the protection of natural resources. The controls or the rupture can identify the transgressions. Authorities grant some individual sanctions to the deviants. Thus, not any collective sanction impacts the respectful farmers of the legislation. The farmers set themselves up in order to increase their productivity and to perpetuate the trust that the state authorities grant (or to avoid the sanctions for everything by having a nearby context in favor of their activity (Wade, 1987).

For the sanctions, the deviants want to exploit randomly the lands being around his exploitations, even those created illegally. At each iteration, the government notes a random number of lands illegally exploited and looks for the one who created the exploitation. After launching several times the simulation with different values of configurations, it appears by observing the variation of the rate of deviants that if the government controls the management's transfers correctly, the sanctions are well applied. The presence of the deviants increases land deterioration. Some deviants however manage to avoid sanctions and run some untaxed incomes. Nevertheless, the application of a sanction to the deviants rebalances the system so that the deviants cannot get any profits from the illegal practice of agriculture or the exploitation of natural resources. Therefore, individual behaviors impact the collective behaviors if a farmer knows the machinations of his neighboring. The farmer generally grants his trust to his neighbor as long as it doesn't involve being a deviant. However, if he knows the acts of deviance of his neighbor, he will be more suspicious against him and will denounce him.

This modelling is quite difficult to implement and require a lot attention. For testing the model, several configurations will be applied to multiple simulations. Usually, these tasks take a lot of time to prepare. Not to waste time, the creation of these initial states should be automated.

3.1.3 MASC with SIEGMAS

In this case study, we are trying to model the interactions between stakeholders in a precise geographical area. To do so we have to create a representation of a region in a simulation platform. In this specific example, the region of Analamanga (Madagascar) which directly usable map data are quite difficult to find, is used. So we go from a map extracted from a digital document that was provided to us (Figure 5).

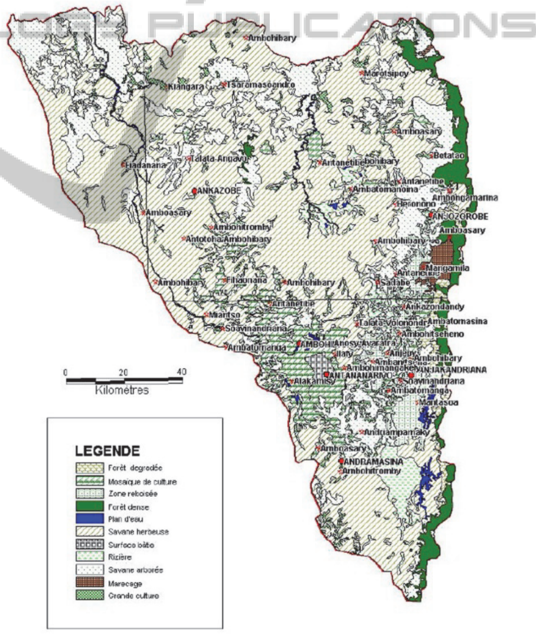

Figure 5: Analamanga region map.

Without MASC, we would have been constraint to crisscross the image in software image processing and then to manually select and assess the color of each sector of the grid. Thanks to MASC, these tasks were simply handled by giving the image file, the number of sectors to cut in the width direction and a color swatches that we simply took from the legend of the map by clicking on it. Afterwards, a preview was displayed to us (Figure 6). We decided then to prioritize the colors of the sectors or not and generate the code snippet regarding the chosen programming language, in this case NetLogo.

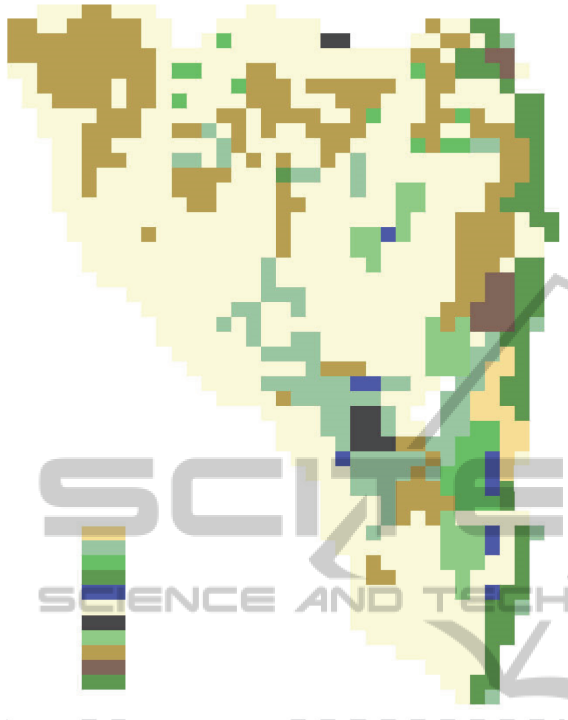

Figure 6: Preview of Analamanga region map generated by MASC.

Part of generated code snippet for Analamanga region map:

```
; Permits to create a representation of
the map
to create-world
    clear-all
    set-patch-size 20
    resize-world 0 39 0 50
    ask patch 0 50 [
        set pcolor [186 158 74
    ]
    ask patch 1 50 [
        set pcolor [186 158 74]
    ]
    ask patch 2 50 [
        set pcolor [255 252 217]
    ]
    ask patch 3 50 [
        set pcolor [255 252 217]
    ]

    [...]

    ask patch 36 0 [
        set pcolor [255 255 255]
    ]
    ask patch 37 0 [
        set pcolor [255 255 255]
    ]
    ask patch 38 0 [
        set pcolor [255 255 255]
    ]
    ask patch 39 0 [
        set pcolor [255 255 255]
    ]
end
```

The advantage of using MASC in this context was to more or less finely crisscross the map to translate from the bitmap file to an exploitable representation of this map inside a simulation platform. As a result we can have easily different grid resolutions of the map with no effort and can deploy rapidly different base maps for different configurations.

This methodology also allows model transposition to different geographical regions, for different simulations executions on Reunion Island and Mauritius regions and comparing obtained results.

3.2 EDMMAS

EDMMAS (Energy Demand Management by Multi-Agent Simulation) is a prototype based on the model of the same name for simulating the production and consummation of energy according to the evolution of the population over the years.

3.2.1 Context

The issue in the EDMASS project is the energy development of a territory using a reproducible modelling. Reunion Island provides a favorable research field in this area thanks to its isolations because its small area (2.500 km²) promotes the quest of new forms of energy development and prohibits electricity production from nuclear power.

EDMMAS is a tool for helping experts to make decisions through ABS, an overview of the evolution of electricity consumption by 2030 and to provide new facilities.

3.2.2 EDMMAS

EDMMAS is and evolution of DS model used to explore various scenarios of land usages and conservation planning that simulates on the whole island, the interactions between the three types of land use (natural, agricultural and urban), in order to allow observation of the changes induced. The energy layer has been added to allow this (Figure 7).

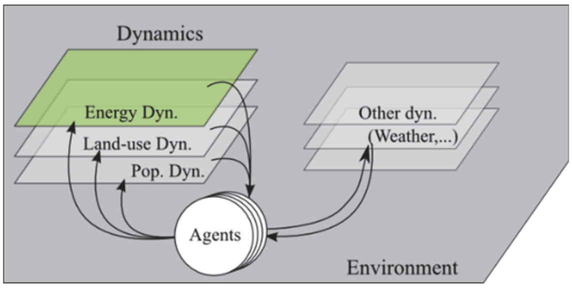

Figure 7: EDMMAS dynamics.

EDMMAS was designed on the GEAMAS-NG simulation platform. This platform has been created with several powerful tools to reduce the gap that lies between the real and complex system model. Among the innovations of this platform, we notice in particular XELOC. XELOC is a support for configuration and initialization of multi-agent systems. It is established on the basis of XML, enriched by a semantic script language. Thus, while a configuration for defining the form of a conventional XML description, XELOC offers the possibility of synthesizing a complex configuration by means of a dynamic process. In addition, XELOC incorporates a rich initialization technique using semantic maps provided by mathematicians. All this makes XELOC, scalable and accessible by nature, a generic support used in many contexts of implementation. A XELOC interpreter has been developed within the GEAMAS-NG platform.

3.2.3 MASC with EDMMAS

In EDMMAS, we handle maps containing information on energy sources available in Reunion Island. We tried to use MASC to complement XELOC Language and facilitate simulation initialization.

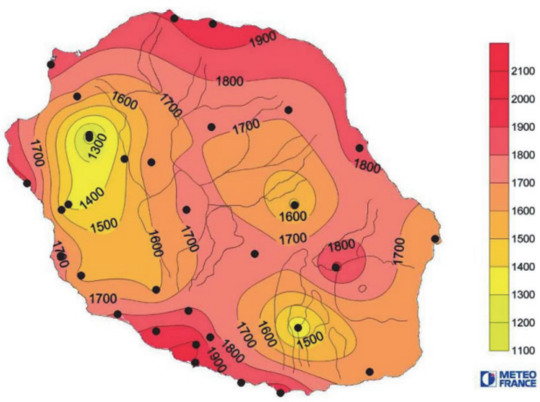

Figure 8: Reunion Island average annual global radiation (2009).

We take a map of average annual global radiation to the horizontal of Reunion Island in 2009 (Figure 8) for the practical case.

Using MASC, we can easily retrieve a XELOC code snippet corresponding to the map with the correct color codes without using multiple tools. When we manipulate codes, we usually don't want to lose time with redundant tasks such as picking all wanted color codes in an image. Thus we benefit of using advantages of MASC and XELOC Language. MASC allows color selection and generates effortlessly code snippets in XELOC language.

Part of generated XELOC code snippet of the example:

```
<forMap name="OurMapName"
      url="./myMapFile.jpg"
      resolution="16">
  <legend name="3classes">
    <label name="rgba(255,18,56)"
           r="255" g="18" b="56" />

     [...]

    <label name="rgba(229,255,0)"
           r="229" g="255" b="0" />
  </legend>
  <cell legend="10classes"
       operator="average"
        mode="closer">
    <case label="level_0">
      <!-- INSTRUCTION FOR #0 -->
    </case>

    [...]

    <case label="level_9">
       <!-- INSTRUCTION FOR #9 -->
    </case>
   </cell>
</forMap>
```

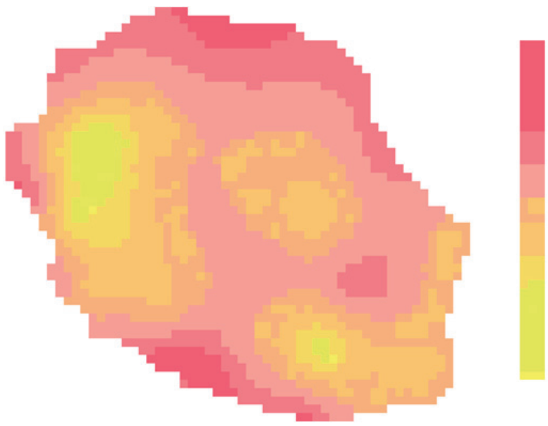

Figure 9: Preview generated by MASC of the average annual global radiation of Reunion Island.

Using MASC for generating XELOC code snippets does not require to send the image on server for a post treatment. In this case, MASC is just an advanced color picker which allows to generate XELOC code snippet as output. We also note that if we need the reuse of this map for NetLogo, we have just to set the grid resolution, ask for extraction, and then change the language choice and regenerate the code snippet that enables us to have an equivalent map in patches for NetLogo (Figure 9).

Part of generated NetLogo code snippet for Reunion Island average annual radiation (2009):

```
;Permits to create a representation of
the map
to create-world
    clear-all
    set-patch-size 20
    resize-world 0 59 0 44

    [...]

    ask patch 19 44 [
        set pcolor [254  91  117]
    ]
    ask patch 20 44 [
        set pcolor [254  91  117]
    ]
    ask patch 21 44 [
        set pcolor [254  122  135]
    ]

    [...]

    ask patch 59 0 [
        set pcolor [255  255  255]
    ]
end
```

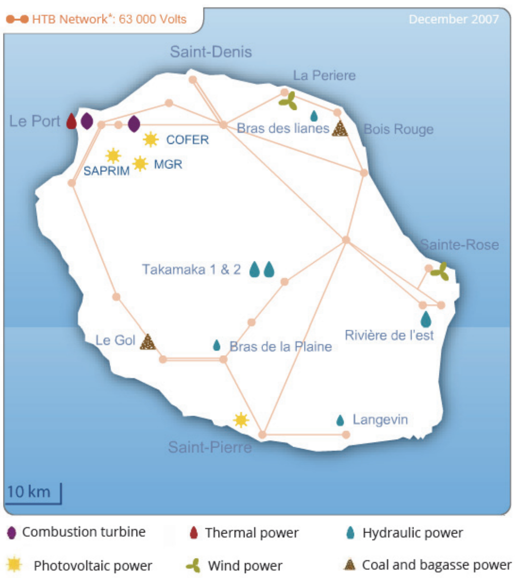

Figure 10: Reunion Island power plants (2007).

We notice in this example of type of use that an advanced version of MASC would be very useful if the tool can also permit to set the location of some agents on the map. In the context of EDMMAS, we can take for example the localization of power plants in Reunion Island (Figure 10) that will be the agents of the model. Keeping in mind that MASC is designed to facilitate the configuration of simulations, one of the future goals of MASC is to allow the placement of agents in addition of creating the environment in which they evolve.

4 FUTURE WORKS

MASC is already a fully functional tool and responds more or less to the initial requirements mentioned in above parts of this paper. However, during our experiencing phases, we note that the tool can be improved to improve its effectiveness.

On the server side, an upgrade of the hosting plan would permit a usage of another programming language for example Java which allows to have a compiled server side application to improve the calculation time instead of an interpreted script. With Java, we could also ignore the passage of data through Apache server and prevent the flow of information between several services. The client side will directly communicate with the server side of the application.

The client side can also be optimized by modifying some functions and not using frameworks as it is now. The advantage of using frameworks is the time gained at developing the tools. Once the proof of concept is acquired, the next step of MASC is optimizations.

Then we can also add more feature on this tool. Actually, the tools only manipulate the space of the simulations. It could also permit to prepare the initialization of multi-agents such as the positioning, setting up clusters of patches or boundaries. We could also expand the list of available multi-agents simulation platform languages to generalize the use of the tool and permit more members of the community of the domain of multi-agent systems to use it.

5 PERSPECTIVES AND CONCLUSIONS

The redundant problem of creating representation of a space in multi-agents systems leads us to find some useful tools to be more productive. We often use maps from GIS softwares, paper supports and other electronic documents that we want to set up in our simulations.

In this article, we presented MASC, a support tool for configuring multi-agent systems based on the information extracted from colors of bitmap images. MASC aims to provide a simplified user-friendly service for the acquisition of code snippets ready to use with multi-agent systems simulations platforms. Thereby, creating multiple configurations for a same modelling can be done quickly and easily, while allowing the users to focus on the data and the results of simulations. Moreover, MASC allows to effortlessly create several of more or less finely crisscrossed map for observing the evolution of agents on different scales and precisions. We could gain more time on configuration if the tool permits also the placement of the agents if needed.

Our purpose in this article was not to bring out the possibility to automate part of the implementation of a modelling, but rather to highlight the lack of user-friendly tools to simplify the initialization of a simulation, which we assessed in example and present to community a more or less adequate solution.

REFERENCES

Brandouy, O., Mathieu, P. and Venryzhenko, I., 2012. Optimal Portfolio Diversification ? A multi-agent ecological competition analysis. *PPAMS (Special Sessions),* pp. 323-331.

David, D., Payet, D., Courdier, R. and Gangat, Y., 2009. XELOC : un support générique pour la configuration et l'initialisation de systèmes multi-agents. *Journées Francophones Systèmes MultiAgents, JFSMA'09.*

Gangat, Y., 2013. *Architecture Agent pour la modélisation et simulation de systèmes complexes multidynamiques : Une approche multicomportementale basée sur le pattern "Agent MVC",* Saint-Denis: s.n.

Gangat, Y., Courdier, R. and Payet, D., 2009. Démonstration : Aménagement énergétique d'un territoire - une approche par simulation multi-agents. *Journées Francophones Systèmes Multi-Agents (JFSMA),* p. 237–240.

Gaudieux, A., Gangat, Y., Kwan, J. and Courdier, R., 2014. Multi-agents model for the study of interactions between the stakeholders in the common pool resources : application to the district of Miarinarivo (Madagascar).

Payet, D., Courdier, R., Sebastien, N. and Ralambondrainy, T., 2006. Environment as support for simplification, reuse and integration of processes in spatial. *MAS. IEEE International Conference on Information Reuse Integration,* p. 127–131.

Wade, R., 1987. The management of common property resources : collective action as an alternative to privatisation or state régulation. *Cambridge Journal of Economics,* Volume 11, pp. 95-106.

Wilensky, U., 1999. NetLogo. Center for Connected Learning and Computer-Based Modeling. *Northwestern University, Evanston, IL.*

APPENDIX

A version of MASC is in public access at http://labs.jkwan.com/masc/.