Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

IMPLEMENTATION OF COMPONENT CONCEPT IN SILENE 2D/3D PRE & POST PROCESSING GUI

Zarko Stankovski
Commissariat à l'Energie Atomique (CEA Saclay)
DEN/DENS/DM2S/SERMA/LTSD
91191 Gif-sur-Yvette Cedex, France
ZStankovski@cea.fr

ABSTRACT

Graphical User Interface Silene is used for generation of input data describing 2D&3D unstructured geometries for neutron transport codes used at CEA. This paper summarizes new capabilities and functionalities of Silene. Silene is designed to deal with unstructured geometries, but can handel regular ones, even they are internally treated as unstructured. Recently the concept of collection of components was introduced. A component is predefined and parameterizable motif. It can be described by a symbolic language. So components can be created, modified and saved using text editor or visual functionalities of Silene. The geometry is generated by assembling all referenced components, contained in the collection.

Key Words: GUI, Unstructured Geometry, Neutron Transport.

1. INTRODUCTION

New numerical methods and algorithms developed as well as the constant growing of computational power allows users of neutron transport codes to solve larger and larger problems. One important aspect of a large problem is the complex reactor geometrie and the difficulties to generate data describing it.

Software's like TDT [1], stand-alone application and module in APOLLO2 [2] code, is a good example. The TDT (Two and Three Dimensional Transport) algorithm solves the Boltzman transport equation using method of characteristic and collision probabilities method. The considered space domains are two-dimensional unstructured geometries, eventually axially expanded on several levels.

The input of data for regular geometries is based on conventions for each type of geometry. Consider a regular rectangular grid, for example. The user should provide two one-dimensional arrays with x and y coordinates and one two-dimensional array describing material distribution. If the grid has diagonal symmetry only one array for the coordinates and a lower diagonal matrix for the materials are required. For each kind of regular geometry programmed in the code, a specific formalism is defined.

The same conception is inapplicable for computer code dealing with unstructured geometries. The meshes of these geometries have arbitrary shapes, so the concept of local description is used by TDT and Silene. There are variants of implementation, but the principle is that one should define line segments and assigned materials (mesh numbers) on each side. To generate "manually" input data describing such geometry, other than very simple one, is virtually impossible.

The Graphical User Interface (GUI) Silene was developed in order to produce input data describing unstructured geometries for TDT with functionalities adjusted for. In the early development stage appeared the evidence that the software should deal with regular geometries also. So the specific functionalities were introduced, allowing regular geometries to be read or generated.

Experience showed that many problems treated are regular or contain regular sub-domains. To facilitate data generation for the large and complex geometries, the concept of components was introduced recently. Component is a parametrable predefined geometry. It can be created visually or described by a symbolic language. Components can be generated, modified, assembled on collections and saved by Silene, with coherence ensured.

Another evolution of Silene was its capability to generate data for codes other than TDT, codes using regular or unstructured geometries. The regular geometries bean treated internally as unstructured. Input data for the codes can be generated from components directly.

2. SILENE GUI

2.1. Implementation

Silene is wholly original software written in Java. "Original" in the sense that the internal data model, functionalities and windows system, was developed especially for dealing with unstructured geometries.

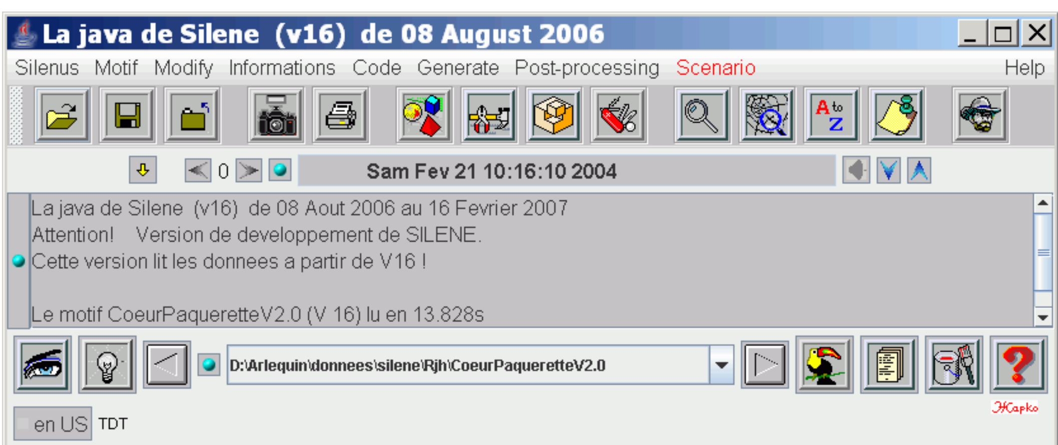


Figure 1. Silene main window.

Initially Silene was written for TDT, with two leading ideas:

- High compatibility with TDT: any geometry generated by Silene can be calculated by TDT and any geometry that can be calculated by TDT should be generated by Silene.

Later developments extended functionalities to generation input data for APOLLO2 modules other that TDT (regular geometries) and for Monte Carlo TRIPOLI4 [4] code.

- Utilizability: a set of qualities that make software to be adopted by the users. It means optimal balance of all usual qualities of any GUI, like ergonomics, easy to learn, easy to use, easy maintenance, robust, reactivity on user requirements etc.

So Silene is designed to be highly user-friendly and easy to operate. Users do not need to learn a new language or complex syntax to use it. It has simple, logical rules with few exceptions. Once the user has understood its principle, he lets himself be guided by menu options and by the construction taking shape on the screen.

2.2. Principle

Motif is the complete set of data describing a geometric construction. Motif may take any shape, provided it is built of connected objects as an unbroken whole. All the informations describing the motif are stored in objects: nodes, equations, meshes etc. Objects are stored in linked lists.

N**odes** (points) contain their coordinates, which are nearly all floating point values in the data model. These nodes can then be used to define four types of **equations**: line **segments**, **arcs** of circle, **circles** and **involutes** of circle.

Sets of equations delimiting closed figures define **meshes**. Meshes are nothing more than closed geometric figures of any shape. **Boundaries** (boundary conditions) are defined in the equations of the motif's perimeter.

In any mesh of a motif, a **reference** can be defined, which is a name of other motif intended to be inlaid in the initial geometry.

Regions are defined by sets of meshes, which may not be adjacent. In APOLLO2 calculations, these regions are the spatial domains in which flux is defined by approximations. **Media** are in turn defined by sets of regions. **Macros** are adjacent regions that are grouped together for "multicell-type" (interface current) calculations in APOLLO2.

Output zones can be defined by sets of regions, essentially used in the post-processing. An output zone can be extracted from a motif to create a new one.

Axially, **levels** (planes) can be defined. The mesh description is the same at all levels, and only regions, macros, media and output zones can be distinguish in 3D.

3. BUILDING ARBYTRARY GEOMETRIES

Arbitrary unstructured geometries are built by drawing the motif on the screen, step by step.

Building starts from a simple predefined figure. The geometry is developed by adding nodes and drawing equations between these nodes. Each time a new equation intersects with another one, Silene adds a new node at the point of intersection and cuts each equation in two; in most cases it also divides the traversed mesh in two. Each time a new mesh is created, a new region is created in the same medium as the original mesh. By adding and removing equations, it is possible to define very complex geometries, such as the RJH core shown on Figure 2.

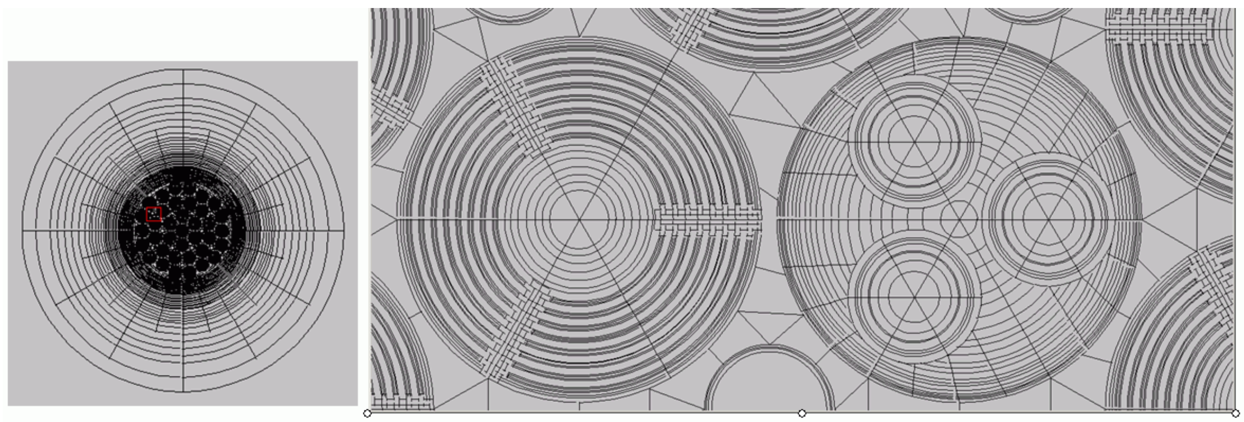


Figure 2. Jules Horowitz Research reactor and zoomed fragment.

The numerous functionalities assist the user on his work. The main menu has 144 items. The **Modify** window manages 159 actions on the objects.

For any object the basic modifications include add, modify, reorder (in the linked list) and delete. Certain functionalities are available in several objects and are applicable under specific conditions, depending on the object. For example: to group objects globally, i.e.: by placing all the regions from two media into a single one while other disappearing. Or to group them partially: by removing one region from a medium and adding it to another.

Actions are designed to minimize the number of entries for numerical values. For example, there are five ways to add nodes: a) at a given distance from a node along one equation, b) by one coordinate, x or y, on an equation, c) by dividing an equation on equal parts, d) at a given distance and angle from a node, and e) by two coordinates.

One very powerful global modification is **Import**: inlay one motif into another. User can create elementary motifs and assemble them. This way the work can be split on elementary tasks and make easy creation of different versions of a complex geometry. There are several other global functionalities, like **Gridding**: globally creating an XY grid in a motif or part of motif, **Join:**

creates a new motif by joining two others, **Extract:** extracts part of motif to create another one, or **Cut:** draws a straight line through geometry to delete part of it.

4. COLECTIONS OF COMPONENTS

Geometry can be defined by a collection of components. A **component** is a predefined and parameterizable motif. It can be described by a symbolic language. So components can be created, manipulated and saved using text editor or visual functionalities of Silene. A component can contain other components, belonging to the **collection**. The geometry is generated by assembling all components referenced in the head component.

Actually Silene deals with ten types of elementary components: **medium**, homogeneous material, **cell**; cylindrical, rectangular and hexagonal, **grid**: rectangular, hexagonal or rectangular of hexagons, **box**: rectangular and hexagonal, and unstructured **motif.** Components can have 3D levels. APENDIX A illustrates all actual components.

Implementation is realized with a great care to simplify and make secure the users work. If a water gap is defined on the edge of a rectangular grid, dimensions of the border cells are automatically adjusted. Hexagonal grid can have a hexagonal edge. Boxes can be nested. Media can have generic names. When a component containing a medium with generic name is inlayed, the name of this medium will be changed, the new one depending of recipient mesh. So placement of a cell with different enrichments can be simplified. 3D levels are automatically adjusted on inlay.

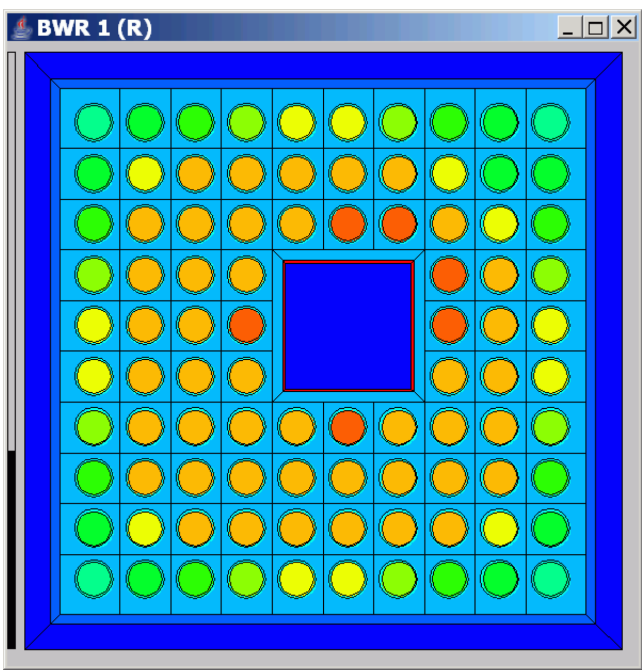


Figure 3. Boiling Water Reactor Assembly.

The components window system is designed to insure friendly and secure work for the user. The BWR assembly of Figure 3 is build with the windows shown on APPENDIX B. Components are listed and managed in the **Collection** window. They can be modified in the specific windows or in the browser. Five different components are used. Rectangular grid **Ass**, defined with two axial levels, is referenced in the assembly box **BWR**. The waterhole box **Box** and two cells **M** (Mox) and **U** (Uranium) are placed in the assembly. While inlaying seven different enrichments (seven media names) are generated. Axial levels off all components are adjusted and the dimensions of border cells are modified to fit the assembly water gap.

Very important feature, not shown in this example, is the capacity to mix regular and unstructured motifs. It allows user refinement of zones in a regular geometry or to simplify generation of an unstructured geometry containing regular zones.

CONCLUSIONS

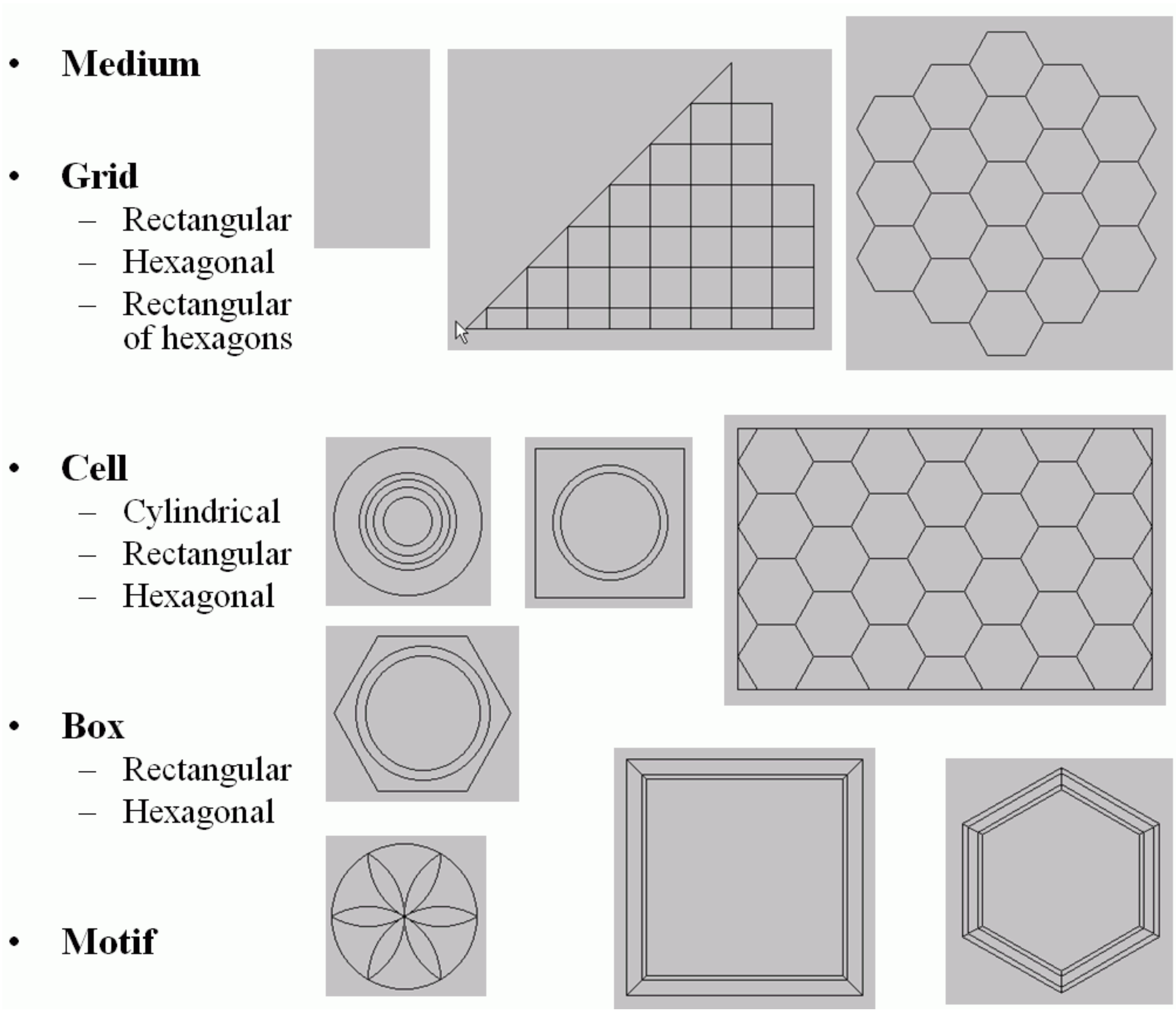
Use of Silene is inevitable for generation of input data for unstructured geometries. Silene was useful for the regular geometries also but often it was easier to produce input data with text editor. In domain of complex regular geometries, implementation of the concept of components allows Silene to be much wildly used: the work is easier and much safer. For large geometries, combinations of regular and unstructured motifs, Silene is still inevitable and now more comfortable to use.

REFERENCES

1. R. SANCHEZ, L. MAO, S. SANTANDREA, “Treatment of Boundary Conditions in Trajectory-Based Deterministic Transport Methods,” *Nucl. Sci. Eng*., **140**, 23-50 (2002).
2. S. Loubiere, R. Sanchez, M. Coste, A. Hebert, Z. Stankovski, C. Van Der Gucht, I. Zmijarevic ,“ APOLLO2 twelve years later,” *M&C 99*, Madrid, September 27 – 30, 1999, Vol. 2, pp.1298-1315 (1999).
3. J.-P. BOTH, Y. PENELIAU, “The Monte Carlo Code TRIPOLI-4 and its first benchmark interpretations. *PHYSOR 96*, p. C 175, Mito, Ibaraki (Japan) Septembre 1996.

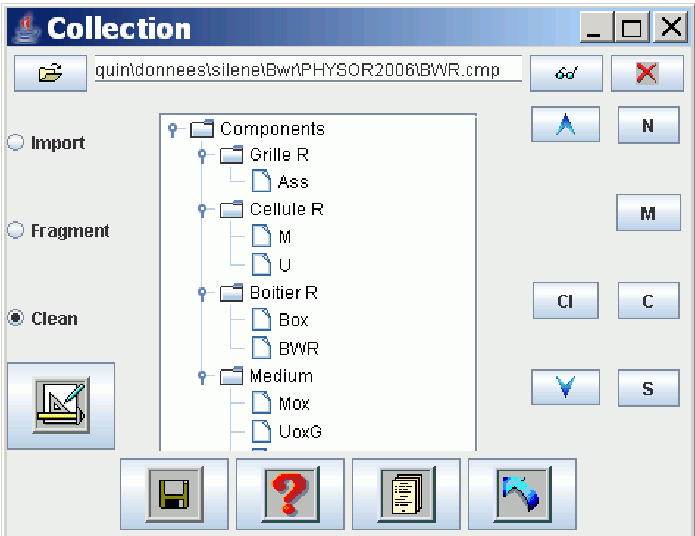
APPENDIX A

COMPONENT TYPES IMPLEMENTED IN SILENE

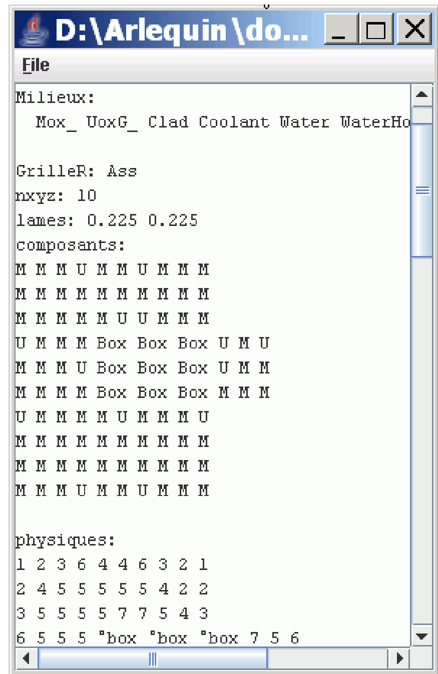


APPENDIX B

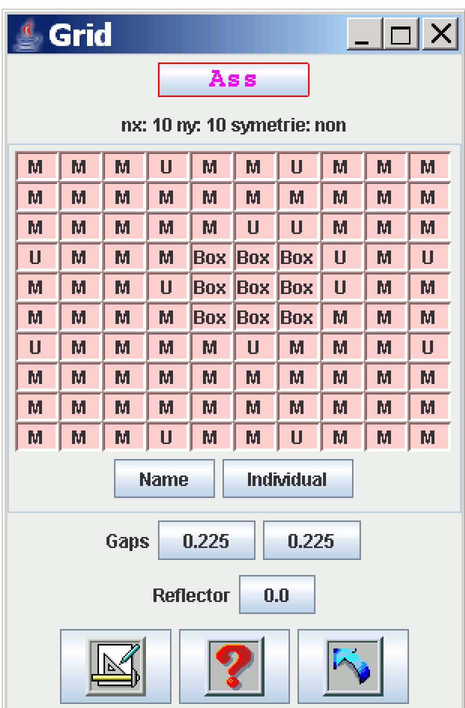
CONSTRUCTING A BWR ASSEMBLY



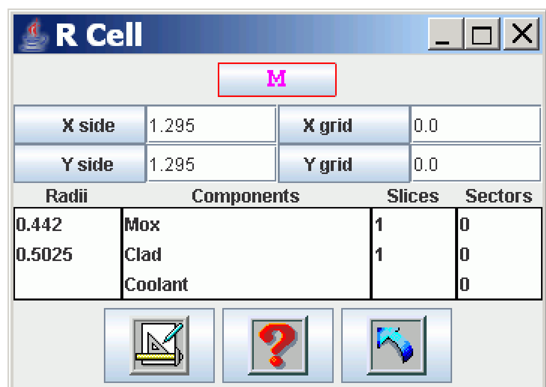
Collection



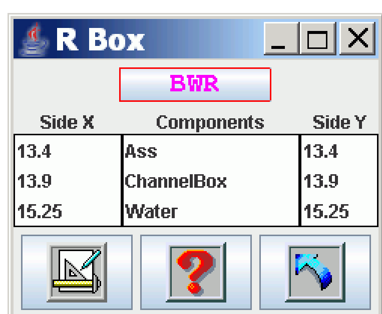
Symbolic language description



Rectangular Grid



Rectangular Cell



Rectangular Box