Documentation: Recalling the forgotten important

John Jairo Huertas Quiroga
Student at European Master Program in System Dynamics EMSD '10
johnjairohuertas@gmail.com

Abstract

From the limited experience expected from an entering practitioner, I found myself in trouble when documenting the results of my participation in a System Dynamics consultancy. While looking for previous work related to documentation at the System Dynamics Review, I realized: (a) no single paper has been fully devoted to discuss how a proper documentation of models should be, and (b) an important fraction of the papers reviewed lacked full documentation of the model and the methodology used to build it, providing few hints into how to document an SD model. Based on existing literature and my experience as a software engineering consultant, a framework that structures a modeling process in terms of phases, workflows, activities, actors and artifacts is introduced as a vehicle to provide supporting information for adequate model documentation. Also the concept of automatic inline documentation generation is introduced to overcome the problems produced by incompatibilities among simulation packages. Four main documents are suggested as the result of the proposed documentation framework. The application of such framework can improve quality of documentation delivered to clients; help building a guide for new practitioners based on accumulated experience and ultimately promote the continued development of the field.

Key words
Documentation, Modeling process, Methodology, Format

Introduction

Which part of all the information gathered during a system dynamics project is/should be documented and delivered to the client? Which phases of the methodology are to be documented in detail? Which structure should the documentation have?

Upon completion of a system dynamics consultancy project recently performed by the author, such questions came to the table. Throughout the project, the consulting team compiled a large amount of notes (some in plain text or in the form of flowcharts), subsystem diagrams, numerical data, interviews, spreadsheet files, managerial presentations and financial statements, just to mention some. The whole process resulted in a final presentation and a documented simulation file complemented by a final report that contained more detailed information about the problem tackled, the reference mode, assumptions taken, variables needing further clarification, description of the tests performed and some recommendations.

The client, who at the end probably was not really sure of what to expect, as a documentation of the modeling process, accepted the set of results and the first phase of the project was considered finalized. All the information collected during the project resides now, either in hard or electronic form, in an archive without any direct connection to the final report. How much of this information should have been formally documented and how? In other words: what level of detail and structure of documentation is enough to guarantee replicability and understanding of a simulation model?

The lack of documentation has been cited as one of the main reasons for model failure or underuse (Gass, 1984; pg. 85). Still, there is no single paper in the entire System Dynamics Review collection dealing specifically with this topic. Only one related paper was found in the proceedings of the System Dynamics Conference of 1983 (Corliss, 1983). Is this because modelers have already mastered documentation activities and no further work needs to be fostered? According to the author's experience and the scarce references related to model documentation found in the literature, this is not the case (Gass, 1984; pg. 84).

Basic concepts in computer models documentation

According to Gass (1979; pg. 5), model documentation is defined as "information recorded during the design, development, and maintenance of computer applications to explain pertinent aspects of a data processing system; including purposes, methods, logic, relationships, capabilities, and limitations". This definition, although oriented to models that needed a lot more of programming efforts to be implemented, gives some hints on critical aspects of modern model documentation: (a) it is recorded during the whole modeling process and, (b) its goal is to provide understanding. Furthermore, the adequacy of model documentation can be assessed if the user can: (a) understand the model, (b) use the model and, (c) keep the model up to date. The latter is particularly important given that system dynamic models are "always in a continuous state of evolution" (Forrester, 1985; pg. 133).

Gass (1979; pg. 14) distinguishes two types of computer model documentation: (a) *inline* documentation which is the documentation entered by the modeler in the simulation package for each model component (the program code) and, (b) *formal* documentation, that includes all documents written by the modeler explicitly to provide adequate documentation.

In the particular context of system dynamics, Sterman (2000; pg. 855) explains why model documentation is important: (a) it allows replicability (giving others the chance to fully replicate the model, check it for errors, understand it and depart from there for further developments), (b) it ensures simulation results can be understood and, (c) it represents a way for the modeler to recall the rationales behind the model structure. Documentation is the first step towards a clear communication of the model (Richardson and Pugh, 1981; pg. 215), and a key to the utility of the model (Gass, 1979; pg. 31).

Model documentation should also be oriented to an audience (Sterman, 2000; pg. 856). This audience may vary, and usually is composed of three different segments (adapted from Gass, 1984; pg. 85; Corliss, 1983): the policy maker interested in the policy recommendations or insights derived from the modeling process (also including optionally the scientific community), the user that operates the computer simulation model, and the modeler itself or another modeler whose goal is to maintain the model over time. Each of these audiences' needs require different views of the same model (Figure 1).

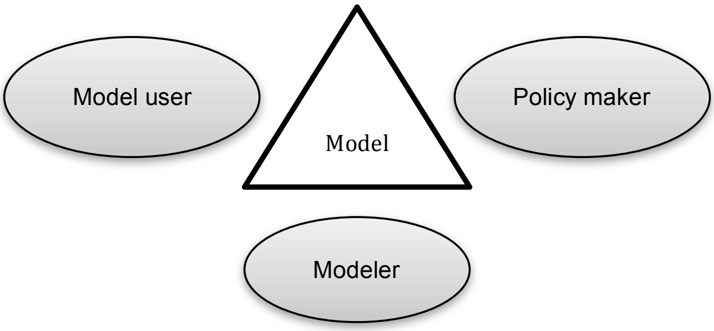

Figure 1. Segments of the audience of a model documentation

Although not abundant, guidelines for system dynamic model documentation have been provided by some leading authors. The following section summarizes the approaches found.

What composes the documentation of a system dynamics model?

Richardson (1996, pg. 147-148) describes two approaches of communicating system dynamics models (Figure 2): (a) a sequence presenting "important stocks and flows, examples of crucial feedback loops, […] followed by a detailed, equation-by-equation explanation/justification" and, (b) "use of sector overviews keyed to model equations and coupled with diagrams detailing important microstructure".

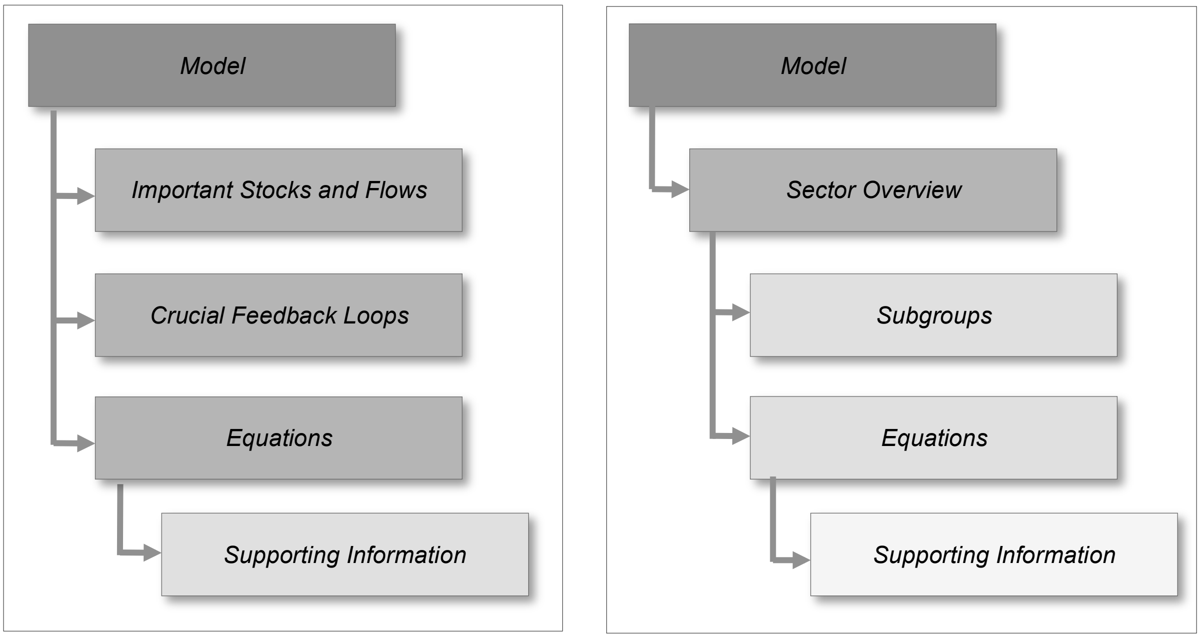

Figure 2. Two approaches on model communication (Richardson,1996)

Sterman (2000; pg. 856) describes in his guidelines what the documentation of a model includes. Figure 3 shows a hierarchical structure where the model is divided in subsystems and for each subsystem its relevant feedback loops are described. Finally each variable is documented with its equations and other supporting information.

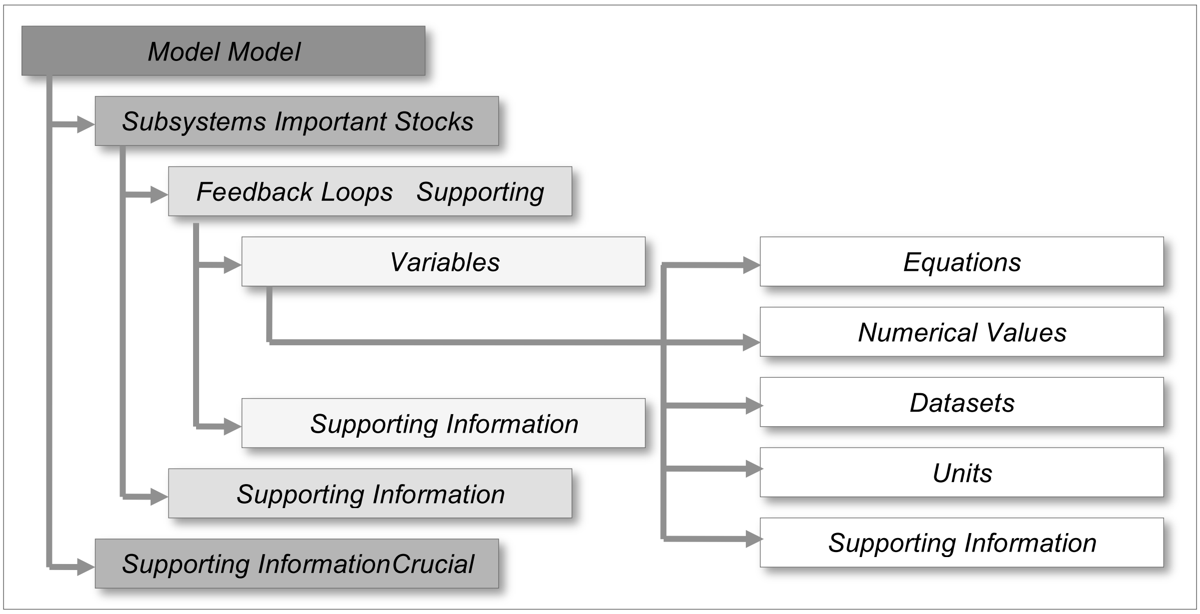

Figure 3. Sterman (2000) approach to model documentation

With these approaches in mind, a review of the latest articles in the System Dynamics Review presenting simulation models was made in order to check which documentation practices are being followed by the academic community or reports of consulting projects.

How is current work being documented?

From a revision of the last issues of the System Dynamics Review it can be concluded that there is no standard practice on how to share the documentation of simulation models, neither of the model itself nor of the modeling process used. Understandably, due to space restrictions most of the times papers present a summarized version of the models with a description of the problem being addressed, a simplified version of the stock and flow or causal loop diagram, model analysis and a description of the results generated by the model.

Besides this summarized version, some papers do not offer a way to get the full documented model (see for example Paich et al, 2011; Bivona and Montemaggiore, 2010; Brady, 2009, Bayer and Gann, 2006). In this case, an email address is provided to ask the author for detailed documentation of the model published. In other cases documentation can be obtained from different alternate sources: (a) by downloading the simulation file using a separate web link (as in Repenning N, 2000; Rahmandadand Hu, 2010), (b) by downloading the simulation file from the supporting information section of the editorial web portal (Rahmandad and Weiss, 2009), and (c) by directly reviewing full stock and flow diagrams and the corresponding equations from appendixes in the paper (Saeed K, 2009). When trying to use any of the previously mentioned means to get access to documentation, the author encountered some problems (see Table 1).

Type of document	Documentation Format	Experience result	Pros	Cons
Paper in the System Dynamics Review presenting a simulation model	Simulation model file	In two cases the file opened with error messages from the simulation package, some equations were missing and the simulation model was not running. In another case the file opened correctly and ran successfully.	No extra work required to review the model in detail or to replicate the simulation results. Equally useful for small and big models.	As simulation packages evolve, they may not support files from earlier versions. The viewer of the file may not have the appropriate software to open the file.
Paper in the System Dynamics Review presenting a simulation model	List of equations as appendix	The model documentation could be reviewed in detail.	Replicable in little time due to its reduced size. No extra work/software needed to access detailed information of the model.	Listing of equations feasible only for small models. Needs extra work to replicate the simulation results in a simulation package.

Table 1. Forms of presentation of the supplementary model documentation, experience results and analysis.

For small models, including the full stock and flow diagram and equations, appendixes proved to be the best documentation approach, since access and replicability are facilitated. In the case of bigger models, providing detailed documentation through a documented simulation file has proven to be error prone. This is because of *incompatibility between simulation packages and even between different versions of the same package*. The incompatibility between versions was already mentioned in Diker and Allen (2006) where they describe it as a challenge to be addressed. In general, *there was no common procedure to access a full documentation and even for the cases when it was reachable, the experience was unsuccessful most of the times*. This not only affects the personal experience of who is directly interested in the model but also does not contribute to raising the quality of the work done in the field. As stated by Sterman (2000; pg. 855): "Documentation is required to ensure that your results can be understood, replicated, criticized, and extended by others […] Demanding that all models be fully documented and all results be fully replicable also defends against add-factoring- the practice of adding a fudge factor to the output of a model so that it squares with the modeler's intuition".

On the other hand the description of the modeling process used to build a model varies in its level of detail, usually presenting the problem definition, the dynamic hypothesis, simulation results, validation and policy recommendations or insights. *The format in which the modeling process is presented is heterogeneous, and usually, unless the main topic of the paper is methodological, little space is devoted to process-related aspects* like: duration of the modeling effort, approximate number of well-defined iterations (if any), roles involved, artifacts[1] generated during each phase of the process (Howick et al, 2006), problems found or lessons learned.

From the last review, in general two main concerns can be highlighted: (a) lack of documentation of the modeling process, and (b) the documentation of the models themselves not being easily accessible. It seems that Richardson's (1996; pg. 148) claims of asking for newer approaches to a "science of model documentation" still apply, bringing up the need for more work on the subject.

Broadening the boundaries of model documentation by including modeling process documentation

The documentation guidelines reviewed before and the documentation approaches used by a sample of articles taken from the System Dynamics Review can be qualified as model-centered[2]. Most of the components provided by these guidelines are aimed to understand the simulation model by analyzing its components. But, is this enough to comprehend the rationales behind the conception and construction of such models? The final form of a model is the cumulative result of all the intermediate products of the process that led to its construction, in a way that true understanding can only be accomplished by the documentation of this accumulation (Gass, 1979; pg. 30). This needs to be evaluated at least from two different perspectives:(a) the consulting client perspective and, (b) the academic audience perspective.

Even though most of the people involved in the client's organization probably is not interested on the details of the modeling process but its final results, some information generated throughout the process can be useful and thereby documenting it is important. From the consulting client perspective, there is also the possibility for a modeler to retake the work done by another modeler as part of a new initiative. In such a case, the new modeler will be certainly interested not only in the model, but also in more detailed information on how the model was built, the lessons learned by the modeler during the whole project and so on. Finally, a newcomer may find it difficult to get to the same rationale used to build the model by looking merely at the simulation file or a final report created based on a model-centered documentation approach. Being able to review how the modeling process was conducted will potentially reveal the artifacts produced in the interim and their contribution to the final result.

From the academic audience perspective, a full documentation of the model and the modeling process would help bringing to written form some of the insights gained by expert modelers throughout their professional practice[3]. The methodological wisdom, reflected in the modeling process rather than in the model or its results, can be revealed only through a structured documentation of the modeling process itself. The lack of a written database of methodological wisdom in the current literature was a concern already presented by Richardson (1996; pg. 148): "New practitioners would find the wisdom and best practice in the field hard to find". As a new practitioner, I can testify this statement is still valid today.

How can the modeling process be then documented? To start, it must be recognized that all modelers tend to think of their different efforts in terms of major phases. Then activities and resources are divided in a per-phase basis so that the project is managed and executed under this categorization. The production of the documentation must be related to each of these stages in such a way that the reader can grasp from the document how the intermediate products led to the final model (Gass, 1979; pg. 31).

Linking the documentation activities to the modeling process also helps the modeler have an explicit association of the modeling and documenting activities, and promotes the practice of documenting from the very beginning, and not waiting until the model is finished when there is no more time or resources left to document properly (Gass, 1979; pg. 31; Sterman, 2000; pg. 855; Corliss, 1983; pg. 9).

Given the dependency between the modeling process and its documentation, creating a structured documentation entails giving structure to the modeling process by: (a) defining which are the phases that compose the modeling process, and once the phases are defined (b) describing for each phase which are the tangible products to be generated and which are the documentation requirements related to them. In the next section a structure is presented to identify the key elements of the modeling process that will serve as a foundation for documentation.

Towards a structured (not restrictive) view of the modeling process as a base for better documentation

A modeling process can be described in different ways. A revision of some classical and recent approaches to the system dynamics modeling process (*Randers, 1980; Richardson and Pugh, 1981;* Vennix, 1996; *Sterman, 2000;* Howick et al., 2006; and Warren, 2008) reveal there is no common criteria on how to describe a modeling process. Although all the approaches listed the compounding activities, only some specified with detail the artifacts to be generated. Some paid special attention to the arrangement of the activities to be executed and all of them used different notations and graphical shapes to visualize the process. In summary, none of them provided a concrete and explicit overview connecting all the stages of the modeling process with the tangible products to be generated.

This lack of agreement and therefore diversity in the level of detail found in the definitions of each of the modeling processes reviewed can have an influence on how familiar the modeler is with the documentation that is to be generated throughout the modeling process. A plain definition that does not outline the activities to be executed and the resulting artifacts, does not allow the modeler to jump into the documentation task intuitively.

A unified structure for describing any modeling process should include all the necessary information a modeler needs to know in order to produce the right documentation, namely: (a) phases of the process, (b) activities to execute on each phase and their proposed order, (c) the iterative nature of the process, (d) roles involved, and (e) artifacts generated from each activity (Rational, 2001). The following sub sections intend to define a structure with these characteristics inspired in the representation of the software development process called Rational Unified Process[4] (Rational, 2001). The structures are followed by simplified examples of the modeling process defined by Randers (1980).

Based on the Rational Unified Process, the structure for defining a modeling process is made up by: (a) process overview and (b) workflow detail (Figure 4).

Process overview

The goal of the "Process overview" is to visualize the whole modeling process described in two dimensions along two axes: (a) the horizontal axis representing the time dimension and "showing the dynamic aspect of the process as it is enacted" (Rational, 2001), it pictures the main phases or stages of the modeling process and the iterations that each phase may require; and (b) the vertical axis representing the content dimension, picturing the workflows (logical sequence of activities) to be executed along the whole process and linking each workflow to the amount of work proposed for each phase from the vertical axis (see top box of Figure 4).

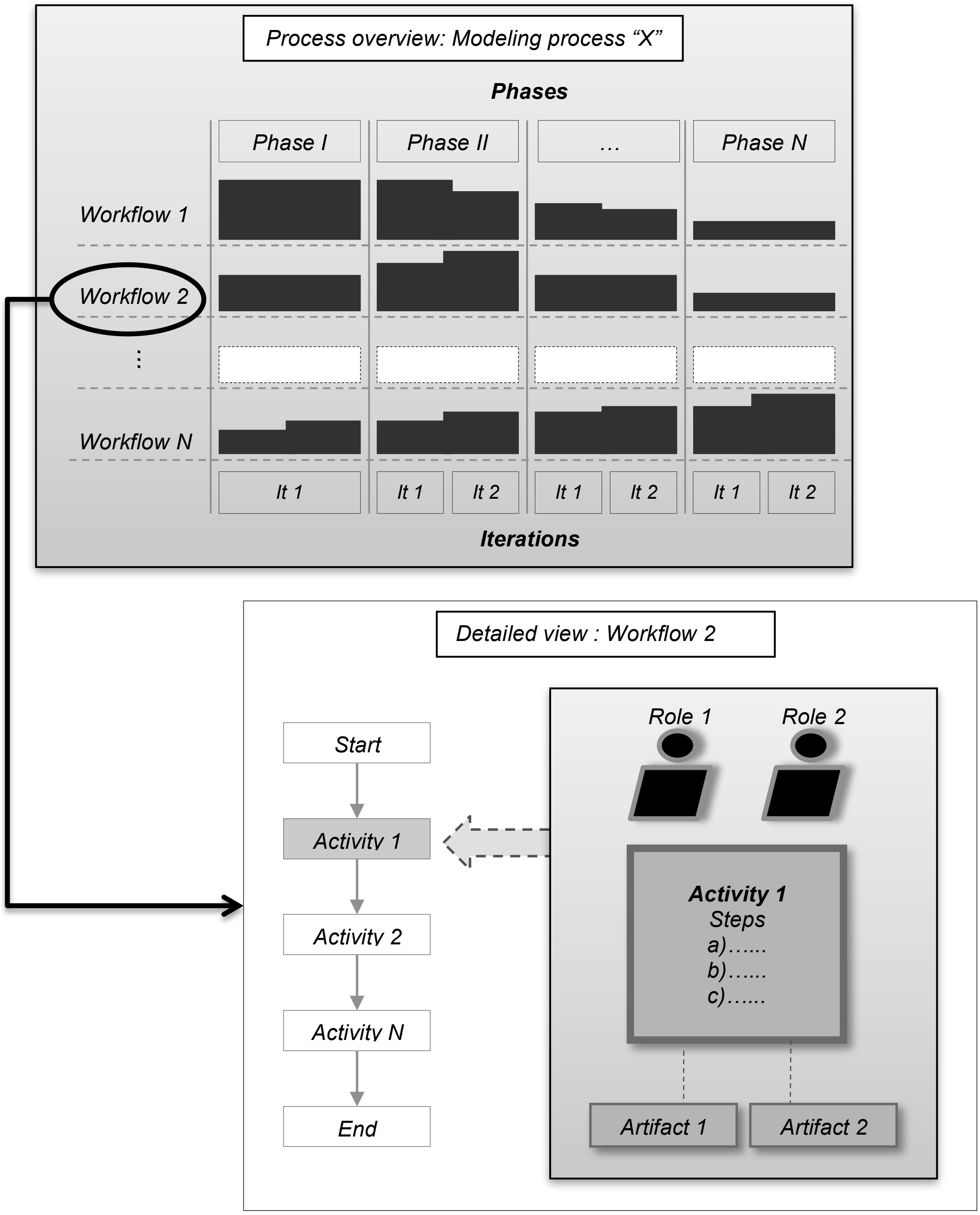

Figure 4. Proposed structure to describe a modeling process outlining information critical for documentation

Phases and iterations (time dimension): As the modeling process is divided into phases or stages, each phase is represented orderly along the time dimension in the upper part of the chart. Each phase may require one or more iterations, represented as subdivisions of each phase in the lower part of the chart. One iteration may on its own require the execution of all the workflows or only some of them. An example of the phases stated in an orderly manner could be as follows (Randers, 1980): (a) Conceptualization, (b) Formulation, (c) Testing, (d) Implementation.

Workflows (content dimension): "A workflow is a sequence of activities that produces a result of observable value" (Rational, 2001). This sequence can be described with a simple flowchart intended to guide the modeler rather than to restrict his activity: "People are not machines, and the workflow cannot be interpreted literally as a program for people, to be followed exactly and mechanically" (Rational, 2001). For Randers (1980) some of the workflows may include: elaborating the reference mode, building an initial model, and running the simulation and policy experiments. A partial visual representation of this process is plotted as example (Figure 5).

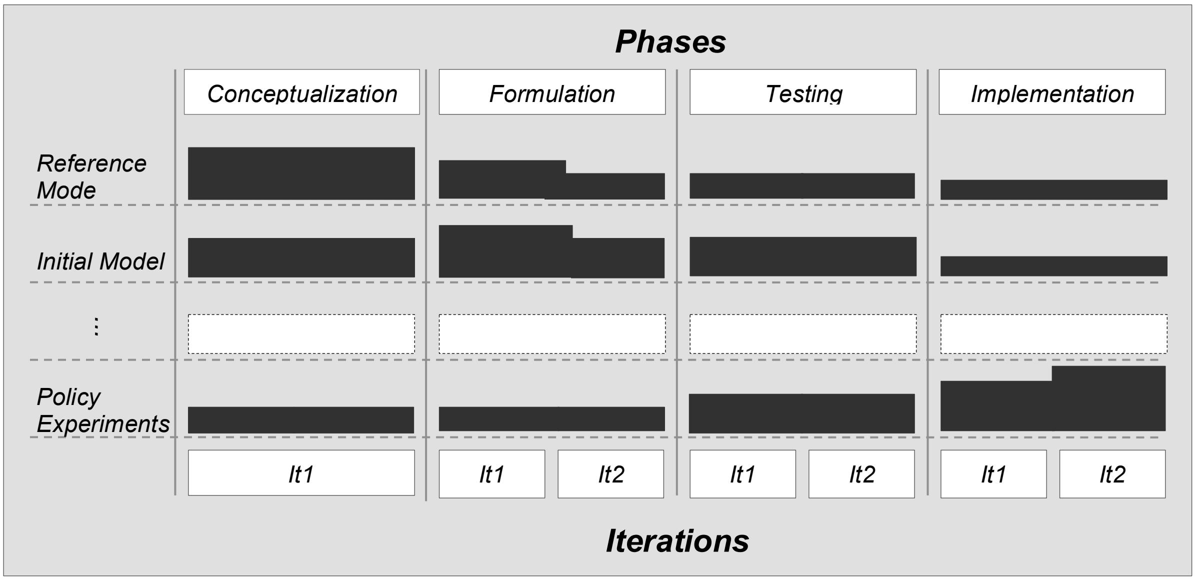

Figure 5. Visual representation of a modeling process

Detailed workflow visualization

Each workflow is then represented in detail by (Rational, 2001): (a) the *roles* that intervene in the activities to execute[5], or the 'who', (b) the *activities* to execute or the 'how' and (c) the *artifacts* to be produced or the 'what'. A visual representation of a detailed description for a workflow is shown (see bottom box of Figure 4).

Role[6]: A role "defines the behavior and responsibilities of an individual, or a group of individuals working together as a team"(Rational, 2001; pg. 8). A role can be thought as "a 'hat' an individual can wear in the project. One individual may wear many different hats"(Rational, 2001; pg. 8). The responsibilities of a role are determined by the activities it is intended to perform and the artifacts for which it is responsible.

Activity (Rational, 2001; pg. 8): Is a unit of work to perform, it must have a clear purpose and specify the artifacts to be created or updated. A single role is responsible for each activity although other roles may participate. Its usual duration should be between few hours to few days.

Artifact: "An artifact is a piece of information that is produced, modified, or used by a process" (Rational, 2001; pg. 9). Artifacts are the tangible products of the process which are elaborated or used while working towards the final product (Rational, 2001; pg. 9). Artifacts may be of different nature: text documents, reference modes, datasets, lists of assumptions, subsystem diagrams, causal loop diagrams, simulation models, lists of equations, tests, policy recommendations, etc. *The sum of all the artifacts composes the full documentation of the modeling project*. In the case of the modeling process proposed by Randers (1980), the 'create initial model' workflow would be represented as in Figure 6:

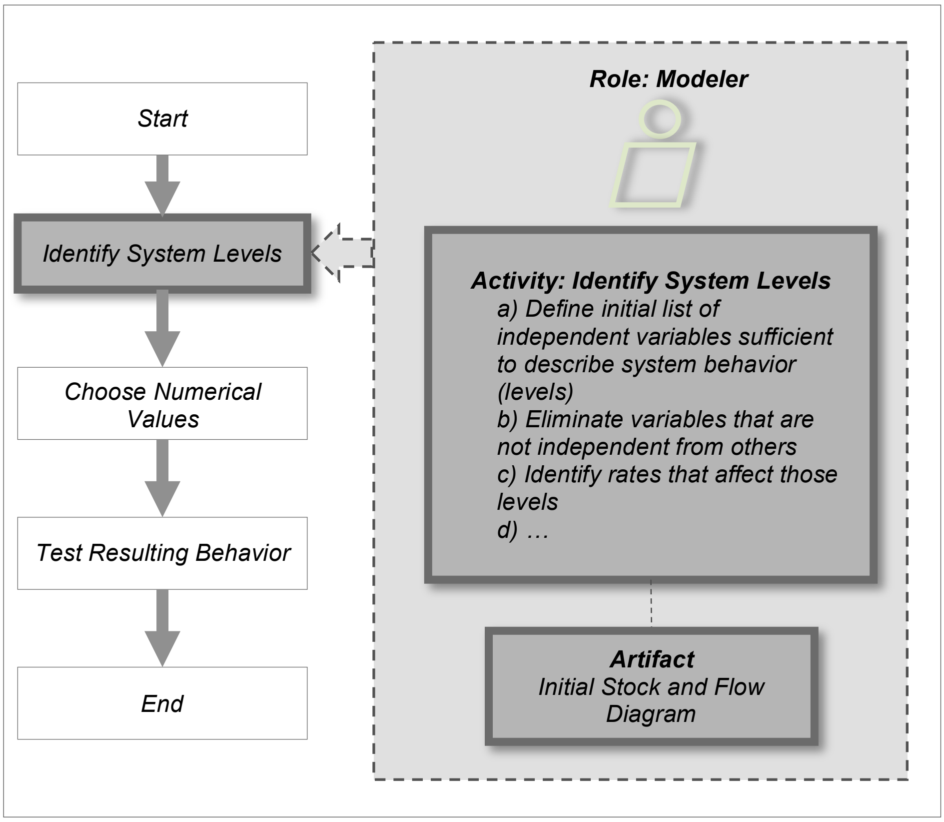

Figure 6. Detailed representation of the workflow to create an initial model

As previously stated, with this structure for defining the modeling processes, the user makes the full documentation of the modeling process explicit and easily traceable to the expected products. Documentation will include the sum of all the artifacts defined in the process[7] and created throughout the modeling project.

A framework for documenting system dynamics models: System Dynamics Documentation Framework

In the previous section, the author pointed out two main issues related to the documentation of the simulation models presented in the System Dynamics Review: (a) little space was given to the documentation of the modeling process and no additional information for that purpose was provided, and (b) when the papers had an accompanying documentation available it came in the form of simulation files, which already proved to be a bad vehicle for sharing documentation given its compatibility problems.

In order to address these issues and give a structure to model documentation, the System Dynamics Documentation Framework is introduced (Figure 7).

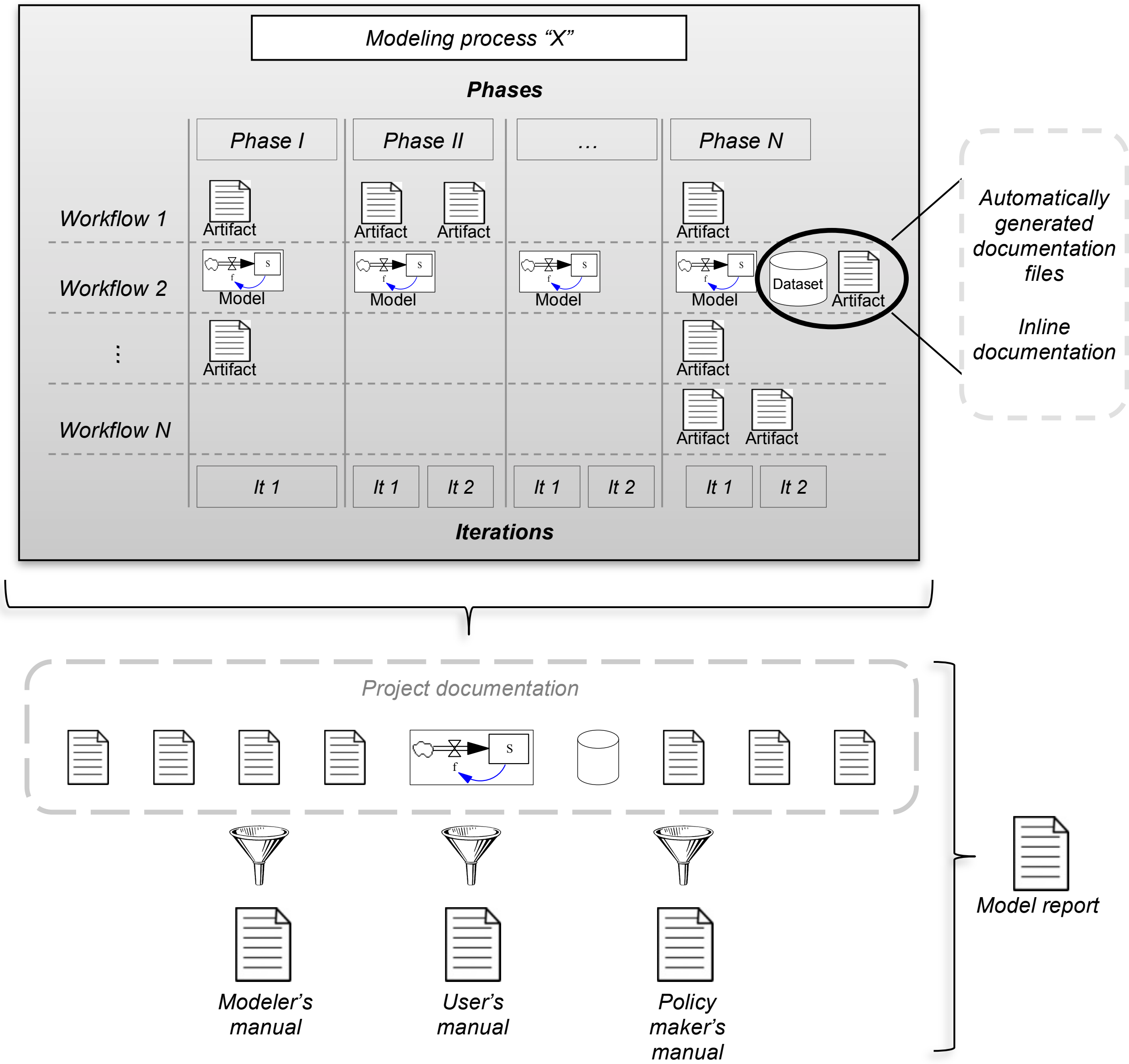

Figure 7. System dynamics documentation framework

This framework provides a view of how a system dynamics model should be documented by linking the modeling process to the artifacts generated in each of its activities. At the same time, the framework provides a general specification format for automatically generated *inline* documentation (generated automatically from the simulation file). The sum of all the artifacts will compose the project documentation. The framework specifies how these artifacts are coalesced in different ways to create four main *formal* documents: *a modeler's manual, a user's manual, a management summary and a model report* (adapted from Gass, 1984; Corliss, 1983). Each of these documents will then serve each of the three audiences identified (Figure 8).

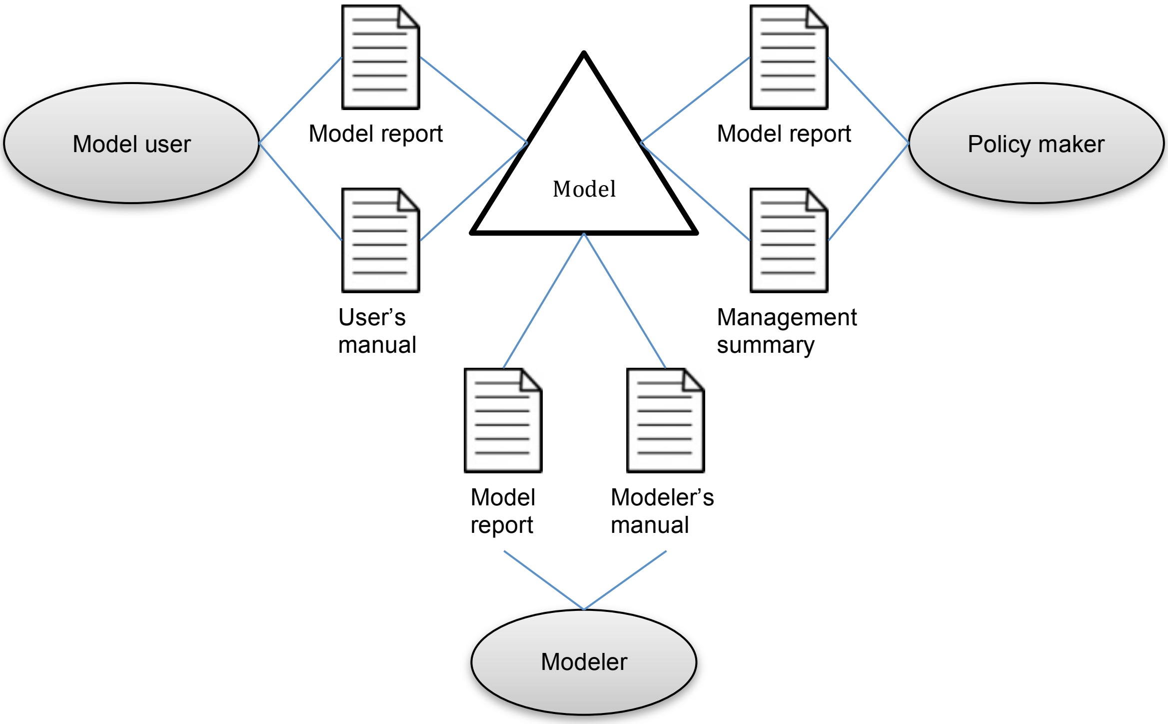

Figure 8. Documents serving the segments of a Model's audience

In the next sections the content of system dynamics model documentation is described, as well as the format for automatically generated documentation from simulation models.

Content of the documentation to be generated by using the framework

The main sources of model documentation are the artifacts generated during the modeling process. Listing these artifacts is not the topic of this paper, as they are used based on each modeler's methodology and thereby would be impossible to have here a comprehensive list. However, as mentioned earlier, listing all the artifacts and linking them to the modeling process is essential for defining the content of the documentation from the beginning. To build this list, we can place the elements of the modeling process in the matrix of phases, workflows, activities, roles and artifacts that was introduced before. If we apply this structure to our modeling process, we will have distinguished the artifacts to generate, first step for building the content of the model documentation.

Although the artifacts are part of the project documentation they are not by themselves its final product. These artifacts are coalesced, combined, and rewritten into four documents, each oriented to a specific audience (adapted from Gass, 1984):

Modeler's manual: Oriented to the modelers audience, combines information from other project artifacts taking care of including "only technical aspects that are essential to practical understanding and application of the model" (Gass, 1984; pg. 87). This manual is a source of reference for those involved in the operation, revision and maintenance of the model. It includes also a description of the modeling process, which could be in terms of phases, workflows, activities, roles and artifacts as presented here. For further detailed reference the reader should be directed to the full artifacts compilation of the project documentation. The modeler's manual is probably the most common documentation product developed; hence it is delivered often in the form of the simulation file or equation listing.

User's manual: Oriented to non-technical users, should provide understanding on the purpose of the model, its mode of usage, its capabilities and its limitations so it can be used effectively. Should specify the different variables of input data in the model (constants and auxiliaries to be modified in the policy exploration process), how the output is presented and interpreted and how to prepare necessary pre-requisites and run the model.

Manager's manual: Oriented to executives and policy makers, this document should include a description of the problem definition, purpose of the model, capabilities, limitations; description of the results generated, its interpretation and use; administrative issues related to the use of the model (i.e. costs, benefits or resources required); role of the model in the organization and decision structure; and basic explanatory material.

Model report: Oriented to any person who needs to determine if the model is of interest, it should summarize, in a concise fashion, basic information about the model.

These four documents plus the project documentation comprise a structured documentation for a system dynamics model. The next section introduces a general specification of how simulation packages should generate automatically inline documentation files and some key characteristics.

Automatically generated inline documentation

Automatically generated inline documentation is a concept that attempts to solve the issue related to lack of accessibility of the simulation model due to incompatibilities in simulation packages. Diker and Allen (2006) already proposed a common interchange language for system dynamics models. Based on XML (W3C-XML) this language would enable the exchange of models between different packages. At the moment of writing this paper, none of the three most known commercial simulation packages support this feature, which leads to the assumption that there are strong barriers against its implementation. Based on this, the concept proposed here focuses on how to generate and share the inline model documentation only in order to make it more accessible and at the same time maintain the proprietary format of each simulation package.

The concept is based on an automatically generated file provided by the simulation tool. The format of the file should be a common open standard agreed by the main simulation software companies. This file should be transformed by the tool itself at least into three formats: a print-friendly format as PDF (Adobe, 2011), a navigation-oriented format as a web page (Richardson, 1996; 148) and a word processor format like RTF (Microsoft Corporation, 2011) in order to facilitate its further modification. The formatting options to be used in these documents should also be part of a standard, in order to maintain visual consistency between the documentation generated by different simulation packages (Figure 9).

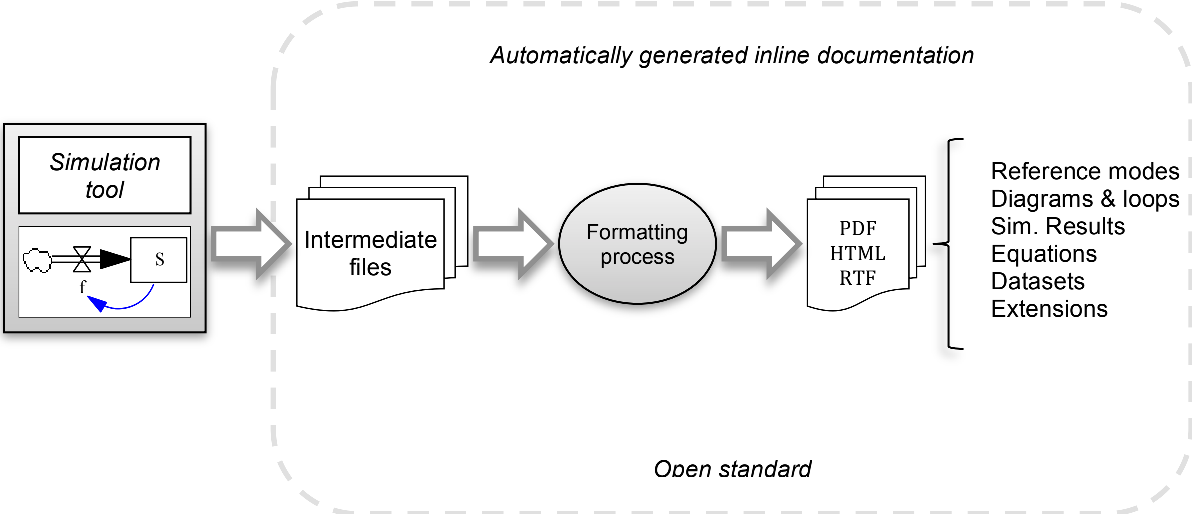

Figure 9. Automatically generated inline documentation

The resulting file should include as minimum the following set of components based on the artifacts described in Sterman (2000; Ch. 3).

Reference modes: The simulation packages should allow the user to select which graphs from the simulation model portrait the reference modes. With this information, the documentation generated in the reference modes section should include all these graphs with its text comments and the supporting datasets to ease its replication.

Diagrams grouped by hierarchical parent and type: The modeler should be able to create a hierarchical single-parent structure of the diagrams built so that high level diagrams, regardless of their type (stock and flow, causal loop diagram, subsystem diagram, etc.) could have many child sub diagrams. These diagrams would be presented according to its hierarchy from parent to children. Documentation for the loops should follow its corresponding diagram when available.

Charts with simulation results: As with the reference modes, simulation packages should allow for the selection of graphs and tables (and any other type of chart) that correspond to simulation results so that these charts are presented together on this section. This simulation results should be grouped by runs (Richardson and Pugh, 1981; pg 213), where a run is defined by: (a) a set of selected parameter values used to generate such results that make part of the adjustments done for testing or for policy exploration purposes, and (b) a textual description. Also each chart should include a text comment of the development of the variables plotted under the run specified.

Model equations and variables: The documentation should include all the equations of the simulation model. The equations should be presented in three listings (adapted from Richardson and Pugh 1981; pg. 213): (a) a list of numbered equations grouped by the sub diagram where the variable's value is set, (b) dictionary of variables and functions in alphabetical order showing the number of the equation where their value is set and its textual documentation, and (c) again a dictionary of variables and functions in alphabetical order, but in this case showing their dependencies, namely the names of the variables that depends on the current variable.

Datasets documented in an import-export friendly fashion: Datasets present in the simulation model should be documented in an import-export-friendly format to facilitate the model replication.

Extension artifacts: In this section each simulation package can embed its own documentation content so additional package-specific features are also documented.

Benefits of the System Dynamics Documentation Framework

The documentation framework could bring benefits to the modeler, the client and the discipline, including:

Methodological guide: Defining guidelines in terms of phases, workflows, activities, roles and artifacts provides a methodological guide, valuable for the new practitioners as a way to adapt best practices already tested by experts.

Put in written form some of the "wise practice" in consulting (Richardson, 1996): New practitioners can take advantage of studying documentation that reveal the modeling experiences of experts in the field in terms of the process used. This may include insights acquired only with the experience and in general consulting wisdom (Richardson, 1996).

Easier communication of the process with the client: By making the process and its partial results explicit, the client has a better idea from the beginning of the modeling process of which guidelines will be taken, which results will arise and who will intervene.

Comprehensive documentation for the client: With this structure the client can trust that the final results of the modeling effort can be traced back to the process that led to them.

Explicit role definition gives an idea of the skills required to conduct a system dynamics process: Although listing precisely which skills are required by a system dynamics practitioner may be a difficult task (Richardson and Andersen, 1995; pg. 130), listing explicitly the different roles a modeler/team should personify would give an idea of which skills are required to do it successfully and how to distribute work inside a team according to their strengths.

Model transparency: Making the model more transparent by showing the history that led to its creation allows for scrutiny, thereby supporting the confidence of its conclusions.

Benefits of the automatic generation of inline documentation

The benefits of documenting the simulation models have been exposed before, the benefits of doing it based on the concept presented here include:

Access to documentation independent from the simulation package: By having a standard documentation format generated from all the simulation packages, it can be assured that the user can review the simulation model independently of the package the model was built with. The generated file can be attached as supporting information of academic papers without facing the compatibility issues of using the package-dependent simulation models but including all the information pertinent to a good documentation[8].

Little extra work to generate a comprehensive documentation: Making the documentation effort centered in the actual process of documenting the model components (*inline*) and not in the generation of a final documentation file is an incentive to the modeler to document from the beginning of the process ensuring that with little extra effort a complete inline documentation will be generated.

Setting a minimum standard for documentation content: The common adoption of a standard of documentation format may push the community to set a minimum content for this documentation. This would also help to establish this standard as an expected product from any modeling effort.

Unified effort to make the documentation better: If all software packages generate a unique documentation format, efforts can be concentrated on this single standard in order to improve it[9].

Conclusions

The lack of literature available about model documentation in general and in the system dynamics literature in particular reveals that further work needs to be done to communicate best practices in the field and establish minimum standards.

It is difficult for a new practitioner to access to full documentation of many of the papers published in the System Dynamics Review.

A description of any system dynamics modeling process in terms of its phases, workflows, activities, roles and artifacts serves as a guide for modelers to produce adequate documentation.

Based on the collection of artifacts that make the project documentation, four documents compose adequate model documentation: (a) user's manual, (b) modeler's manual, (c) management summary and (d) Modeler's report.

The adoption of an open standard to produce inline documentation files from the simulation package would help to provide better access to documentation from any system dynamic model in general and from the models described in the System Dynamics Review in particular.

Further work is required in a detailed definition of the framework and concepts presented in this document.

Summary of reactions and thoughts after the presentation at the International System Dynamics Conference 2011

Taking into account that the goal of the paper is not to provide a deeper academic knowledge about a topic but to initiate a movement around better practices in model documentation, I considered relevant to include some of the reactions I received after the presentation and some thoughts about future lines of work.

After the presentation there were many opinions among the audience on the need to document the modeling process, the lack of functionalities available in the actual modeling software that would allow modelers to do so, and the need for commitment of software developers in improving such shortcomings. Also, I got positive opinions suggesting the incorporation of extra documenting features into the simulation packages that could facilitate the automatic generation of a comprehensive documentation file (containing reference modes, dynamic hypotheses, validation tests, etc.)

I attended a presentation by Ignacio Martinez-Moyano about his System Dynamics Model Documentation and Assessment Tool (SDM-Doc). In essence, this tool takes the model file from Vensim and generates an automatic documentation in HTML format. One point is worth highlighting: (a) it allows the user to include in the comments field of each variable certain metadata (key words), so it generates the documentation according to the instructions written by the user in the comments field using such metadata. This approach has a great potential as it could allow the system dynamics community to create a metadata specification standard from where more structured information could be registered in the model's comments and later translated into a particular format.

From this presentation a new idea came to my mind to start working on a better documentation approach without putting all the work on the software makers: (a) to create a documentation metadata standard so key words can provide structured information about each variable/model component, (b) to create a standard XML format for model documentation, and (c) to ask the software makers to generate a standard XML file containing all the information (including metadata) registered by the user in the comments section. Furthermore, with all the open source movement around the world, it is probable that soon there may be groups of people working on templates to transform this XML files into HTML navigable files, PDF files and other formats. With so many more pending tasks in the "science of documentation" this would be a good start for creating a clearer trace of the System Dynamics work.

Notes

---

[1] The term artifact as described here refers to different tangible products result of the modeling process, some may be input to further stages or final results. Examples include literature reviews, causal loop diagrams, interviews, scenario maps, simulation models and report between others.

[2] From this point of view, documenting a model by solely delivering the simulation model file would comprise the inline documentation but would leave out the formal documentation.

[3]Richardson (1996; pg. 143) includes in his list: wisdom about problem definition and system conceptualization, wisdom about building confidence in models for policy analysis and wisdom about model-based consulting practice from the decades of experience of the great system dynamics consulting firms.

[4] Also known in the software industry as the RUP, since its popularization in the 1990's established a common language for description of the software development process from which other methodologies have departed to establish their own approach.

[5] The classic approaches to the modeling process as summarized in Luna-Reyes and Andersen (2003) do not make an explicit reference to different roles that could participate from the modeling side in a modeling process (contrary to the client side). In this case a single role of modeler could be used for the whole process representation.

[6] The original name for this item in Rational (2001; pg. 8) is worker. However the name role is more appropriate for its purpose in the system dynamics modeling context as it suggested by the same source: "in the Unified Process the worker is more the role defining how the individuals should carry out the work".

[7]Formal models are by themselves difficult to communicate to a broad audience (Richardson, 1996; pg. 147), this issue is not subject of this paper although may be indirectly addressed through a better documentation.

[8] Richardson and Pugh (1981; pg. 213) lists the desired properties of a good model documentation.

[9] A community of different commercial actors (including competitors) working together towards the improvement of standards already proved to be successful in the information technology industry, an example is the standard for the next HTML version (W3C, 2011), created by a group called WHATWG (Web Hypertext Application Technology Working Group, 2011), where competitors in the industry of web browsers collaborate to improve the user experience on the web.

Acknowledgements

I am deeply grateful to Maria Franco for all her assistance in reviewing and providing suggestions for this paper.

References

Adobe. Adobe Portable Document Format, http://www.adobe.com/products/acrobat/adobepdf.html / [1 March 2011]

Andersen and Richardson. Scripts for group model building. System Dynamics Review (1997) vol. 13 (2) pp. 107-129.

Bayer and Gann. Balancing work: bidding strategies and workload dynamics in a project-based professional service organisation. System Dynamics Review (2006) vol. 22 (3) pp. 185-211

Bivona and Montemaggiore. Understanding short- and long-term implications of “myopic” fleet maintenance policies: a system dynamics application to a city bus company. System Dynamics Review (2010) vol. 26 (3) pp. 195-215

Brady M. Advertising effectiveness and spillover: simulating strategic interaction using advertising. System Dynamics Review (2009) vol. 25 (4) pp. 281-307

Corliss J. Model documentation: Hinderance or help. Proceedings of the 1983 Intl. System Dynamics Conference, Chestnut Hill, MA: System Dynamics Society.

Coyle G. The practice of system dynamics: milestones, lessons and ideas from 30 years experience. System Dynamics Review (1997) vol. 14 (4) pp. 343-365

Diker and Allen. XMILE: towards an XML interchange language for system dynamics models. System Dynamics Review (2006) vol. 21 (4) pp. 351-359

Ford A. (1999) Modeling the environment, Island Press, Washington, DC.

Forrester. “The” model versus a modeling ‘‘process”. System Dynamics Review (1985) vol. 1 (1) pp. 133-134.

Gass S. Computer model documentation: a review and an approach. U.S. Dept. of Commerce, National Bureau of Standards. 1979.

Gass. Documenting a Computer Based Model. Interfaces (1984) vol. 14 (3) pp. 84-93

Homer J. Reply to Jay Forrester's “System dynamics—the next fifty years”. System Dynamics Review (2008) vol. 23 (4) pp. 465-467

Howick et al. Linking event thinking with structural thinking: methods to improve client value in projects. System Dynamics Review (2006) vol. 22 (2) pp. 113-140

Luna-Reyes and Andersen. Collecting and analyzing qualitative data for system dynamics: methods and models. System Dynamics Review (2004) vol. 19 (4) pp. 271-296

Microsoft Corporation. Rich Text Format (RTF) Specification Version 1.9.1.http://www.microsoft.com/ [1 March 2011]

PaichM. et al. Pharmaceutical market dynamics and strategic planning: a system dynamics perspective. System Dynamics Review (2011) vol. 27 (1) pp. 47-63

Rahmandad Hand Hu K. Modeling the rework cycle: capturing multiple defects per task. System Dynamics Review (2010) vol. 26 (4) pp. 291-315

Rahmandad and Weiss. Dynamics of concurrent software development. System Dynamics Review (2009) vol. 25 (3) pp. 224-249

Randers J. 1980.Guidelines for Model Conceptualization.In*Elements of the System Dynamics Method*, ed. J. Randers. Cambridge, Mass.: MIT Press. Reprinted byProductivity Press, Portland, OR.

Rational. Rational Unified Process - Best Practices for Software Development Teams. Rational Software White Paper TP026B, Rev 11/01 (2001) pp. 1-21

Repenning N. A dynamic model of resource allocation in multi-project research and development systems. System Dynamics Review (2000) vol. 16 (3) pp. 173-212

Richardson G. Problems for the future of system dynamics. System Dynamics Review (1996) vol. 12 (2) pp. 141-157

Richardson,G.P., and A. L. Pugh III.1981.Introduction To System Dynamics Modeling With DYNAMO. Portland, OR: Productivity Press.

Richardson and Andersen. Teamwork in group model building. System Dynamics Review (1995) vol. 11 (2) pp. 113-137

Saeed K. Can trend forecasting improve stability in supply chains? A response to Forrester's challenge in Appendix L of Industrial Dynamics. System Dynamics Review (2009) vol. 25 (1) pp. 63-78

Sterman, J.D. (2000) Business Dynamics: Systems Thinking and Modeling for a Complex World', McGraw-Hill Companies, Boston, Massachusetts.

Vennix JAM. 1996. Group Model Building: Facilitating Team Learning Using System Dynamics. Wiley: Chichester.

W3C. XML, http://www.w3.org/XML/ [9 November 2005].

W3C. HTML, http://www.w3.org/HTML/ [1March 2011].

Warren K. 2008. Strategic Management Dynamics. Wiley: Chichester.