

TEMPER : **a Toolbox for Engineering Models to Predict Explosive Reactions**

Emmanuel LAPÉBIE, DGA/DET/CEG
33 (0)5 65 10 56 54, emmanuel.lapebie@dga.defense.gouv.fr

Frédéric PEUGEOT, NATO/MSIAC,
(32 2) 707 54 47, f.peugeot@hq.nato.int

Abstract

TEMPER is a "toolbox" of empirical and analytical models dedicated to ammunition safety. Experts can use it in order to aid in the prediction of the response of a munition to a mechanical or a thermal threat. TEMPER main features are:

- A simple Graphic User Interface (GUI) enabling, from dedicated libraries:
 - A direct selection of the scenario (Stimulus / Mitigation / Structure) to be assessed.
 - A direct selection of the model to be run.
- An automatic compatibility management between the chosen model and the assessed scenario.
- An ability to perform parametric or stochastic simulations by varying one or two parameters of the problem.
- An ability to draw curves and save results using an embedded Excel workbook.
- The possibility to have multiple developers.

This software is made available by France/DGA/CEG to MSIAC nations experts as an "open source", Object-Oriented Programming code in order to allow easy integration of custom models or the enhancement of existing ones.

The objective of this project is to support the use of modeling as the primary method for assessing munitions safety. The ultimate goal is to provide to the community a common tool that could become a reference in the S³ community if models and parameters exchanges between participants to this project go well.

1 HISTORICAL BACKGROUND

1.1 VAMPyr story at ETBS

The history of TEMPER begins with that of its ancestor, VAMPyr (Validation des Anti-Mines Pyrotechniques), a tool developed by Frederic Peugeot and Emmanuel Lapébie, when both were working at the Bourges Technical Establishment (ETBS – Établissement Technique de Bourges).

VAMPyr was intended to validate projectile and shaped charge jet mine neutralization systems, using a criterion for the shock initiation of explosives developed by Peugeot during its PhD^{1,2,3}. It was also applied successfully to other problems, such as ERA initiation⁴.

VAMPyr, although innovative in some aspects, was mainly the "hard-coded" implementation of a single model, and it was impossible to simply plug new models or new aggressions in the code without rewriting it completely. Furthermore, the processing of the results was mainly done by hand. Its development ceased at ETBS with the transfers of its two "core developers", Frederic Peugeot to NATO/NIMIC (now MSIAC) and Emmanuel Lapébie to DGA/CEG.

1.2 CEG switched to (a good) TEMPER

CEG (Centre d'Études de Gramat), among other missions, is the French leading center for the physics of high explosives, with dedicated test facilities and a high level of expertise in numerical modeling. Because CEG works also on vulnerability and lethality assessment codes, which includes many empirically fitted or analytical models, it was definitely a good place to start the development of a code dealing with the response of energetic materials to a wide range of stimuli.

At CEG, Emmanuel LAPEBIE wrote a new code, initially called VAMPyr 2003⁵ *in memoriam* of the good old days. Since there are many more features in it, and not a single line of code in common with VAMPyr - ETBS, and also to avoid confusion, it was decided that this software tool deserved a new name: TEMPER, which stands for Toolbox of Engineering Models for the Prediction of Explosive Reactions.

As will be detailed in the next paragraphs, TEMPER is no more a single-model code, but rather a *library* of stimuli, mitigation devices (anything in the real world able to convert an initial stimulus into a "residual" stimulus), explosive-containing structures and models.

¹ Peugeot F. ; « Étude de la vulnérabilité de matériaux énergétiques à l'agression par jet de charge creuse (Study of the vulnerability of high-energy materials to shaped charge jets) » ; PhD at the University of Poitiers ; 1997.

² Peugeot F. ; « An analytical extension of the critical energy criterion used to predict bare explosive response to jet attack », Propellants, Explosives, Pyrotechnics, volume 23, pages 117-122, 1998.

³ Peugeot F. ; « Threshold energy: a dimensional extension of the critical energy for jet attack application », Propellants, Explosives, Pyrotechnics, volume 23, pages 188-192, 1998.

⁴ Lapébie E. ; « How Reactive is an Explosive Reactive Armor ? » ; 20th International Symposium on Ballistics ; pp. 1020-1027 ; Orlando (USA) ; 2002.

⁵ Lapébie E., NOLD O. ; « Développements Récents du code VAMPyr » ; EuroPyro 2003 ; pp. 741-754 ; Saint-Malo (France) ; 2003.

1.3 Cooperation with MSIAC (formerly NIMIC)

In-house developed semi-empirical and empirical codes have advantages and disadvantages. Among the advantages is the fit to the user's need; among the disadvantages are the time (and therefore the money) spent to develop it and usually the absence of timelessness usually because there is no user's guide and no record of the modifications made to the code.

Because MSIAC has been supporting the use of modeling as the primary method for assessing munitions safety for a long time, and because MSIAC is in an ideal position, as an Information Analysis Center, to become a focal point for exchanging/sharing engineering models within the MSIAC member nations' experts, MSIAC accepted CEG proposal in 2004 to make the TEMPER code available to the experts of MSIAC member nations with also a view to ease its international development.

2 Using TEMPER

2.1 User's interface

The Graphic User Interface is made of three windows :

- The "input" window (Figure 1), allowing the user to select the stimulus, the mitigation, and the structure containing the Energetic Material (EM) in accordance with the scenario that needs to be assessed ; to select the model to be used ; and also to create and delete files,
- The "simulation parameters" window (Figure 2), enabling to launch a unitary, a parametrical or a statistical calculation,
- The "output" window (Figure 3), presenting and exploiting the results (embedded Excel file) with both graphs and data sheets (threshold curves, simple curves and data points).

2.2 Running a simulation

A simulation starts from the main window. From this window (Figure 1), the following selections are required:

- A stimulus (spherical fragment, rising temperature, etc) & the data file describing the stimulus;
- A mitigation (air, single layer, etc) & the data file describing the mitigation;
- A structure (cylindrical explosive, bare flat explosive, etc) & the data file describing the structure.

The texts of the selected data files appear in the text boxes situated below the drop-down lists. As said previously, the user is able to create new files ("New" buttons) or to delete files ("Delete" buttons) for each category (stimulus, mitigation, structure).

Once the scenario has been described, a model needs to be selected. Because not any stimulus can interact with a given mitigation or a given structure, and vice-versa, an automatic management of the compatibility between assessed configurations and models has been added in the code. As an example, Table 1 gives all the allowed possible combinations for the James critical energy model⁶.

⁶ JAMES H. R. ; « An Extension to the Walker-Wasley criterion for the initiation of solid explosives by projectile impact » ; AWE Report n° O 21/82 ; 1982.

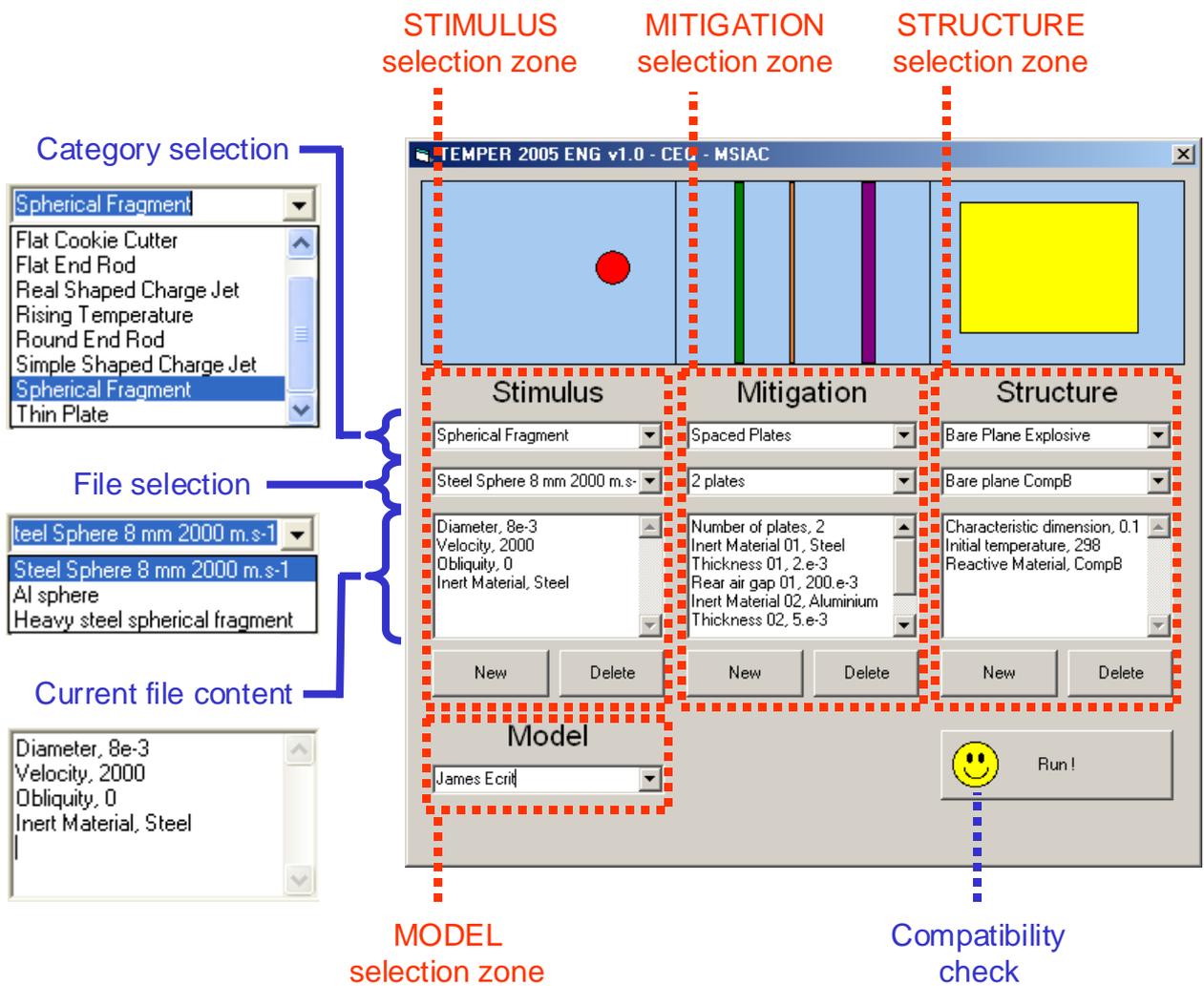


Figure 1: Input window

Table 1: James critical energy criterion : declared compatibilities in TEMPER v1.0

James Critical Energy model		
Stimuli	Mitigations	Structures
Flat end rod	Air	Bare plane explosive
Round end rod		
Spherical fragment	Spaced plates	
Spherical fragment		
Simple shaped charge jet	Air	
Real shaped charge jet	Single layer	
Real shaped charge jet		
Thin plate	Air	
Flat cookie cutter		

Once the configuration has been described and a compatible model selected, TEMPER offers the choice of a unitary, a parametrical or a statistical simulation as shown Figure 2 :

- Unitary simulation : all the variables describing the configuration are fixed.
- Parametric simulation : the user can choose to vary one or two variables between a minimum and a maximum value, with a constant increment (coefficient = 1), an increasing increment (coefficient > 1), or a decreasing increment (coefficient < 1).
- Statistical simulation : allows variation of one or two variables in accordance with a probabilistic law. The current possibilities are the normal law (the user enters mean and standard deviation) and the uniform law (the user gives the minimum and maximum limits of the distribution). The user also specifies the number of random points to be picked in order to reproduce the statistical dispersion of the variable.

Note that it is possible to select a parametric simulation for the first variable and a statistical simulation for the second. The variables themselves are automatically exported by the current stimulus, mitigation and structure (see Part 3 for a description of “objects” in TEMPER).

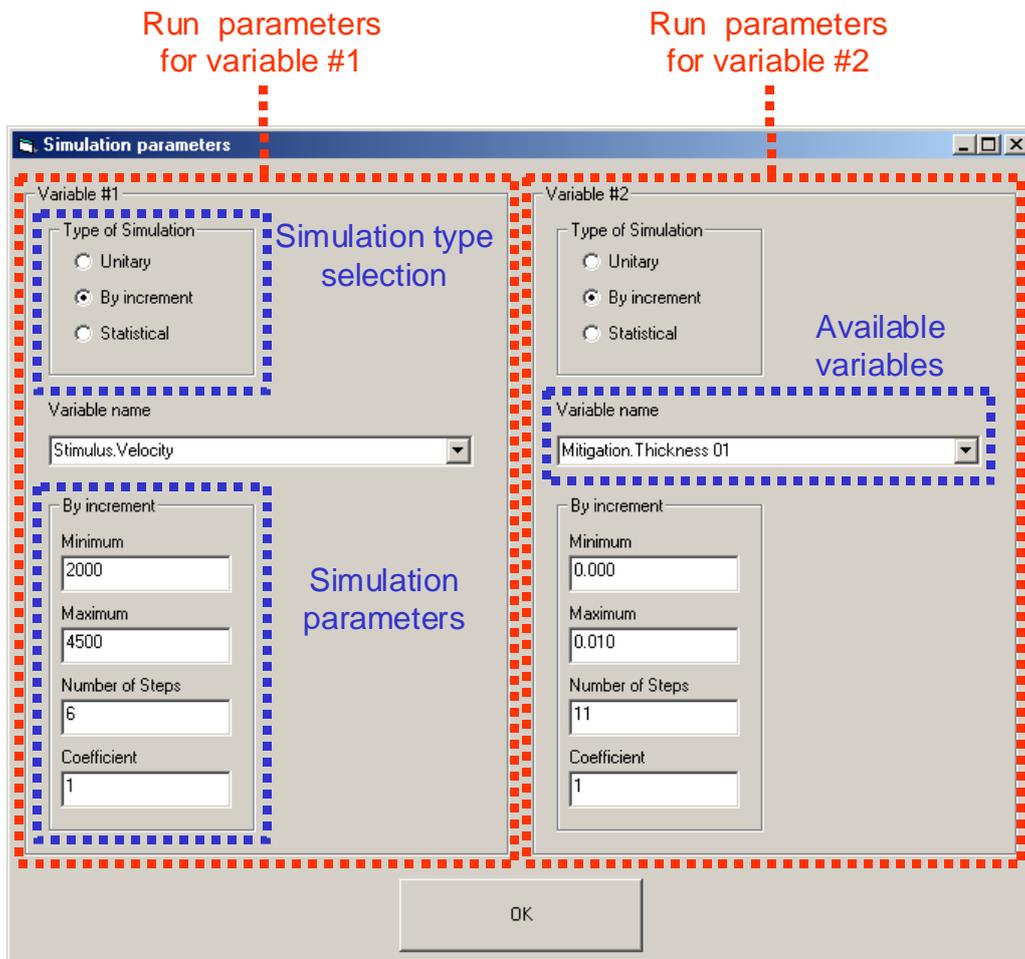


Figure 2: Simulation parameters window

Once the calculation is run, results are automatically stored, exported to an Excel worksheet and presented through the output window. An example result from a parametric simulation is given Figure 3.

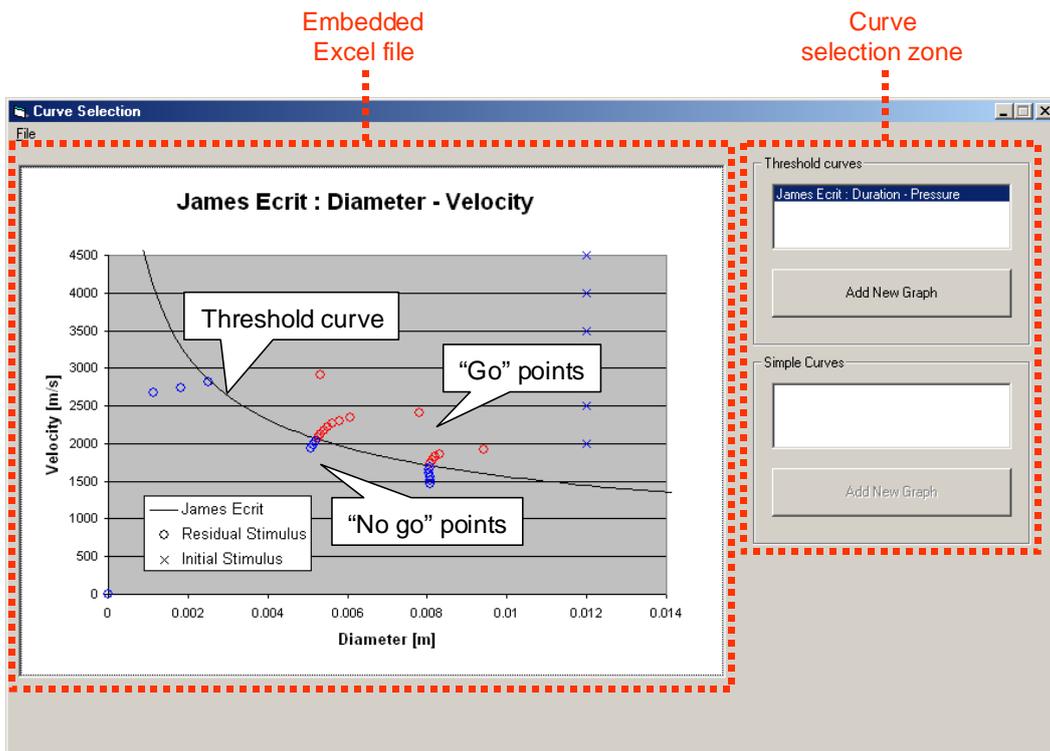


Figure 3: Output window, Threshold curve

During unitary simulations, simple curves can also be issued by either the stimulus, the mitigation or the structure taking part in the scenario. For instance, the “Spaced Plates” mitigation uses Thor laws to compute the loss of mass and velocity of a spherical fragment through each plate. Such curves could also be drawn and viewed in the embedded Excel workbook, as shown Figure 4.

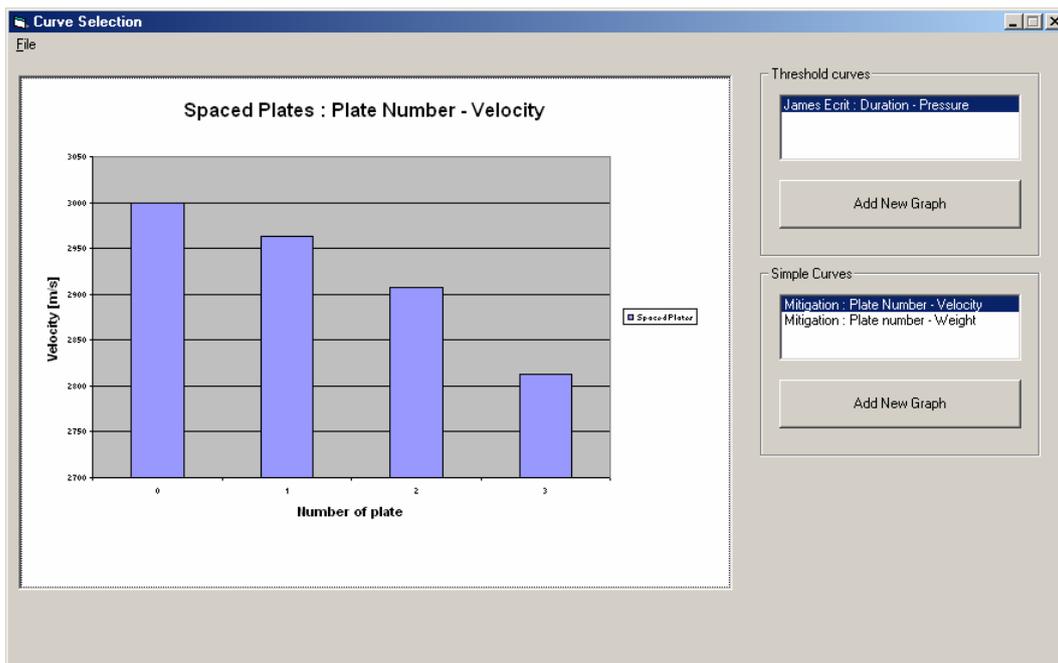


Figure 4: Output window, Simple curve

3 Developing TEMPER

The main driver of this project is to be able to have multiple developers in different organizations at the same time. By doing so, it is possible to share development costs, avoid duplication efforts and rationalize the modeling activity with the goal to have a fully recognized international tool.

3.1 Programming language

The basic idea in coding TEMPER was to make very simple the addition of new models. The (not so) easy way to do this was to choose an object-oriented programming language, much more modular than classical ones (such as Fortran).

Microsoft Visual Basic 6.0 (VB) was chosen for the many advantages it possesses when compared to its disadvantages (see) and especially because it is an easy-access, widely used environment with a friendly Graphical User Interface (GUI).

Table 2: VB advantages and disadvantages

Advantages	Disadvantages
<ul style="list-style-type: none">• One of the most frequently used languages.• Is structured and has powerful functions.• Incorporates a graphic interface (event language).• Object incorporation functions are particularly advanced.• "Object" capabilities are sufficient, although incomplete	<ul style="list-style-type: none">• Not fully object-oriented,• Development is only done under Windows environment.• Deployment of executables is manageable but rather complex.• Executables must be accompanied by associated files (OCX controls, etc).• Slow execution.

A platform-independent programming language, such as Java, might have been a better choice, even though Java programming is less easy, but Java was not so widely used at the beginning of the project.

3.2 Objects ?

Each "entity" in the real world is seen as an "object" by the code. Objects are an extension of structured types (or "records"), able to store not only different kind of variables, but also functions and subroutines.

In TEMPER, the main "objects" are grouped in four categories:

- Stimuli (which may trigger a pyrotechnic reaction).
- Mitigations (which are able to transform the initial stimulus into a residual stimulus).
- Structures (which contain an energetic material liable to react).
- Models (describing the reaction of a structure to a stimulus).

Table 3 lists the objects available within TEMPER-ENG-v1.0.

Table 3: TEMPER ENG-v1.0 objects

STIMULI	MITIGATIONS	STRUCTURES	MODELS
Flat end rod	Air	Bare plane explosive	V ² d (Held)
Round end rod	Spaced plates	Bare cylindrical explosive	u ² d (Held)
Flat cookie-cutter	Single layer	Bare spherical explosive	E _{crit} Walker - Wasley
Spherical fragment		Covered plane explosive	E _{crit} James
Simple shaped charge jet			Y (Yactor)
Real shaped charge jet			V _{threshold} (Jacobs - Roslund)
Thin plate			t _{cook-off} (Creighton - Victor)
Constant Temperature			
Rising Temperature			

Basically, all the work is done by objects, and that is why the “main” code of TEMPER is so light. Moreover, objects of the same category share the same subroutines, and a simple copy/paste is really half the job required to add a new object.

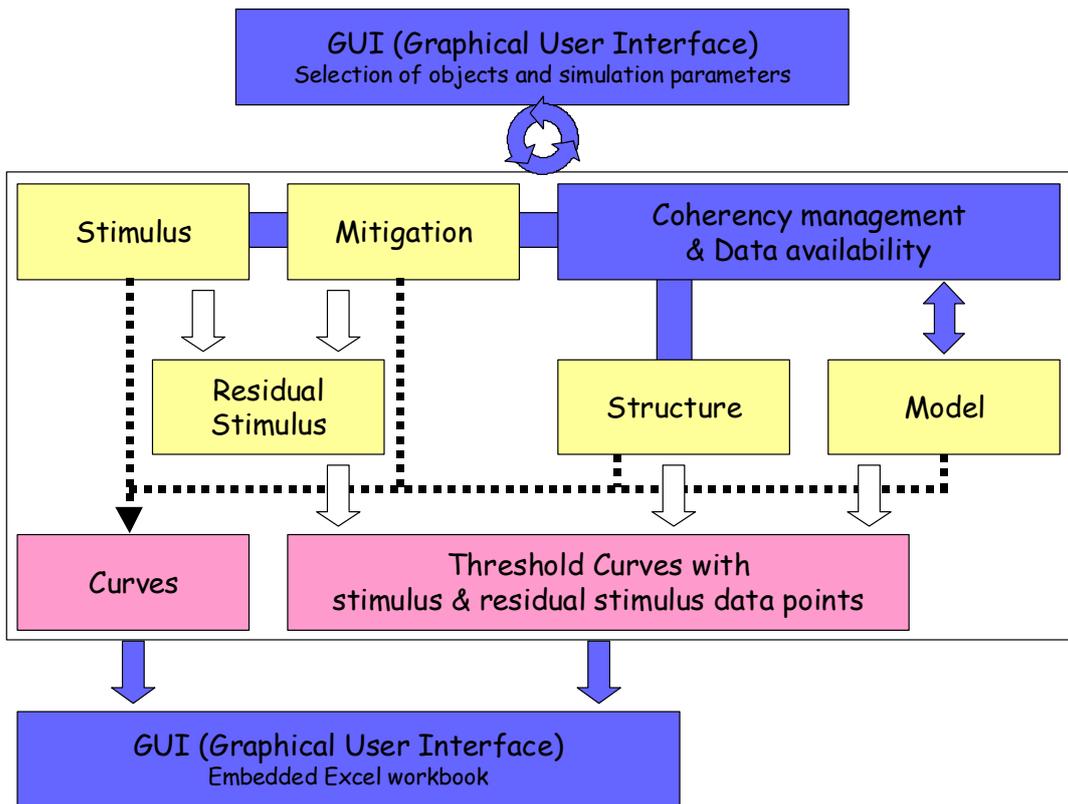


Figure 5: Main objects in TEMPER

Objects interact to mimic real-life events (see Figure 5) : if a spherical fragment ("Stimulus") interacts physically with a covered high explosive ("Structure") protected by spaced thin plates ("Mitigation"), the code will ask the "Mitigation" object to “virtually” interact with the "Stimulus" object in order to provide a residual "Stimulus" (another object).

Then, the "Model" object (this time not a "physical" object, but a mathematical description of the reaction thresholds, for example Held's v^2d criterion or Walker and Wasley critical energy) will receive both the residual "Stimulus" and the "Structure" objects and creates several new objects (among other things, "Threshold Curves" that will be drawn in the output window).

Without going too deeply in details, objects have many other useful properties. For instance, they automatically export the variables appearing in the "simulation parameters" window, so there is no need to rewrite the code each time we add an object. The same thing happens in the "output" window : the list of available curves from the "Stimulus", "Mitigation", "Structure" and "Model" objects is requested from the objects themselves.

Detailed information about the hierarchy of objects, their properties (variables) and methods (subroutines and functions) can be found in the users/developers guide⁷, available from the password protected MSIAC website (see Part 4).

3.3 Incorporating new objects

It is not the purpose of this document to present in details the procedure to incorporate new objects. Such a feature can also be found in the users/developers guide provided with the documentation package associated to TEMPER. Nevertheless, a developer who wishes to incorporate a new stimulus, mitigation or structure object, or even to code a new model, can do it very simply with minimum modifications to the main code. As an indication, the incorporation of a relatively simple empirical or analytical model requires about one day of work.

4 TEMPER availability

TEMPER specificity is that users and developers are neither within the same location nor within the same organization. In order to achieve the ultimate goal to provide to the community a common tool that could become a reference in the S^3 community if models and parameters exchanges between participants to this project go well, it was decided to create a TEMPER e-working group with MSIAC as a focal point.

Experts interested in the project can participate using the web at two levels:

- The users level, i.e. experts that use TEMPER, provide the group with some feedback regarding the quality control of the product, the range of validity of the implemented models, new parameters to share with the others but also with new ideas/requirements for development.

- The developer level, i.e. super users that also use this platform to develop models for private usage, share the newly implemented models (i.e. programming) but not necessarily the parameters coming with. It should be noted that TEMPER has specially be written as an Object Oriented Programming code in order to ease its development by multiple developers. Basically the coding of a new empirical model can take less than a day because everything but the specific features of the model is standard. Reading or writing to files, assessing internal variables, automatic post processing coding, the possibility to perform statistical calculations for example does not require any added line of code. Developers could well be students supervised by users and who would be given the task to implement a model. This would be an incredibly fast way to promote and ease the development of TEMPER.

⁷ Lapébie E ., « TEMPER 2004 : Utilisation et Développement », Rapport 2004-012359/CEG/DT/EX.

From the password protected MSIAC website, the TEMPER e-working group's participants can download:

- Executables and source codes,
- Documentation associated with the quality control of TEMPER
- Documentation associated with the use and development of TEMPER

4.1 Executables and source codes

Executable Code TEMPER v1.0 For declared "Users Only"		Open Visual Basic 6.0 Source Code For declared "Possible Developers"	
 English Version	 French Version	 English Version	 French Version
✓	✓	Available Soon	Available Soon

For obvious reasons, it should be noted that only the English version will be maintained in the future.

4.2 Documentation associated with the quality control of TEMPER

- Bugs Reports (BR). As software bugs are discovered, users will be able to submit a detailed BR, which will be escalated to a software developer for review and resolution;
- Request For Support (RFS). The RFS Tracker will be used mostly to track usability problems and usage questions related to the operation of the software that require an immediate response. A Frequently Asked Questions section will probably be made available by then;
- Request For Feature (RFF). The RFF Tracker will allow users to submit requests for enhancement (RFE) to the software. Submissions may be closed when the requested changes are incorporated into a patch of the software, or when rejected by developers;
- Patches (P). As probably only one version of the software will be available every year and as developers will have had the possibility in the mean time to incorporate new models in their own version, the P Tracker will enable participants to have an up to date list of capabilities proposed by TEMPER with developers details for using the Patches.

4.3 Documentation associated with the use of TEMPER

- Users/Developers Manual (UDM) in French and in English.
- A tutorial enables users to get used to the principal features of TEMPER.
- THOR equations (edition 2): provides information and parameters on THOR model.
- Jacobs-Roslund (JR) model (edition 2): provides information and parameters on JR model.

4.4 Documentation associated with the development of TEMPER

- Users/Developers Manual (UDM) in French and in English.

- New energetic material datasheet implementation procedure: example of implementation of a new energetic material datasheet for existing models.
- New model implementation procedure: example of implementation of a new energetic material datasheet for existing models.

5 TEMPER v1.x and beyond

5.1 Temper v1.x

As said previously, sharing TEMPER development between different organizations is a method to make good progress in a short time. Multi-developers development already began with:

- The implementation of Peugeot Model for fragment and shaped charge impact. This includes also the Peugeot threshold energy criterion.
- The implementation of a one on one and a one on many sympathetic reaction stimuli enabling sympathetic detonation to be predicted.

NEW STIMULI	NEW MODEL
One on One Warhead	Peugeot
One on Many Warhead	Peugeot threshold energy criterion

This patch will be made available to TEMPER users under the denomination TEMPER v1.1. The documentation supporting TEMPER users has been updated accordingly with 2 reports:

- Peugeot model (edition 1): provides information and parameters on Peugeot model.
- Critical energy models (edition 1): provides information and parameters on Walker & Wasley critical energy criterion, James and Peugeot modified critical energy criteria and Peugeot threshold energy criterion.
- Additionally, in order to help determining the shock sensitivity constants required to run the Peugeot model, a specific excel worksheet has been prepared for TEMPER users. This worksheet not only contains spreadsheets enabling these constants to be calculated automatically from plate impact or pop plot experiments, but also a Pop Plot coefficients database (28 reported), a critical diameter database (107 reported) and a Hugoniot parameters for inert material database (32 reported).

5.2 New features

On the programming side, current object hierarchy is not perfect and minor changes will be made in the next version in order to fulfill object-oriented programming requirements. A part of the code has been developed during training periods and should be rewritten (creation and deletion of objects, and to a lesser extent Excel links, does not respect “object-orientation”).

Concerning new features, here is a non-exhaustive list of what could be done:

- Manage the compatibility between the Stimulus, the Mitigation, the Structure and the Model by using the validity domain of the model. For example, the Thor

model parameters used by the "Spaced plates" Mitigation object are only valid for steel a fragment, which is not checked yet. Likewise, the Jacobs-Roslund parameters are not valid for all projectile or covering materials. This is based for the moment on the expertise of the user but it could be made automatic.

- Make US or SI units available at the user's discretion.
- Improve the management of the parameters of models. This is managed for the moment by models but it could be managed by energetic materials (a single file for each EM, incorporating all model data)
- Propose reverse engineering possibility by optimizing automatically the parameters of a model to fit the tests results (for example loading a list of stimuli leading to a positive result and a list of stimuli leading to a negative result, and automatically determine the threshold curve and the associated best parameters between the two sets of points.

5.3 New objects and models

Concerning new objects and models, it will be up to the developers to program them. Because of its object-orientated programming, any kind of model can be used or called by TEMPER, so it is theoretically possible (although probably useless!) to couple TEMPER with a 3D hydrocode !

There are a myriad possibilities, with no limits but the imagination of the user. TEMPER even lends itself to uses far from the reactivity of high-energy materials, such as the penetration of rods into complex targets, the dispersion of atmospheric pollutants, etc.

6 CONCLUSION

TEMPER is a "toolbox" of empirical and analytical models dedicated to the prediction of the response of a munition to a mechanical or a thermal threat. Its main features are:

- Simple Graphic User Interface (GUI) enabling, from dedicated libraries:
 - A direct selection of the scenario (Stimulus / Mitigation / Structure) to be assessed.
 - A direct selection of the model to be run.
- Automatic compatibility management between the chosen model and the assessed scenario;
- Ability to perform parametric or stochastic simulations by varying one or two parameters of the problem;
- Ability to draw curves and save results using an embedded Excel workbook.
- Multiple developers

This software is made available by France/DGA/CEG to MSIAC nations experts. In order to facilitate access to this software, a TEMPER e-working group with MSIAC as a focal point has been created. Participation to this project is simply on a demand basis.

The ultimate goal of this project is to provide to the community a common tool that could become a reference in the S3 community if models and parameters exchanges between participants to this project go well.