







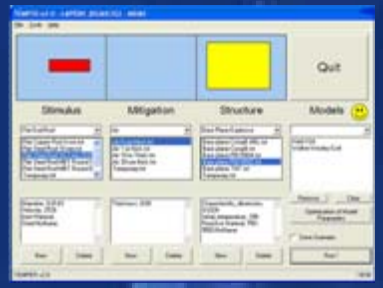



TEMPER : 2009 development roadmap

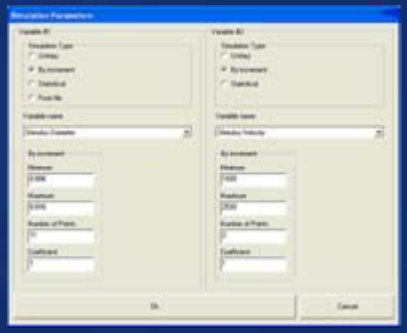
E. Lapébie (DGA/CEG)
N. Daly (SME/CRB)
P.-F. Péron (NATO/MSIAC)

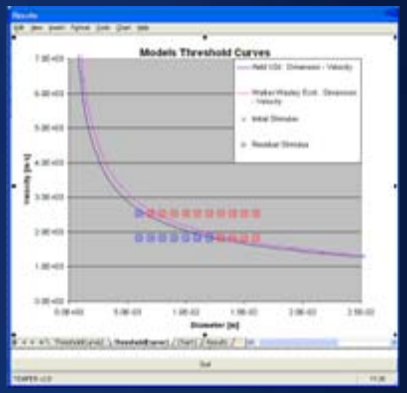
What's TEMPER ?

- **Overview**
  - TEMPER, an acronym for Toolbox for Engineering Models to Predict Explosive Reactions, is a "library" of models dedicated to ammunition safety.
  - TEMPER, developed by DGA/CEG, has been made available through NATO/MSIAC to experts from MSIAC member nations.
  - Since the first version in 2004, many improvements have been brought about. The current version (TEMPER v2.0) has been released in October 2007.
- **Using TEMPER in 3 steps**



SCENARIO SELECTION


SIMULATION PARAMETERS



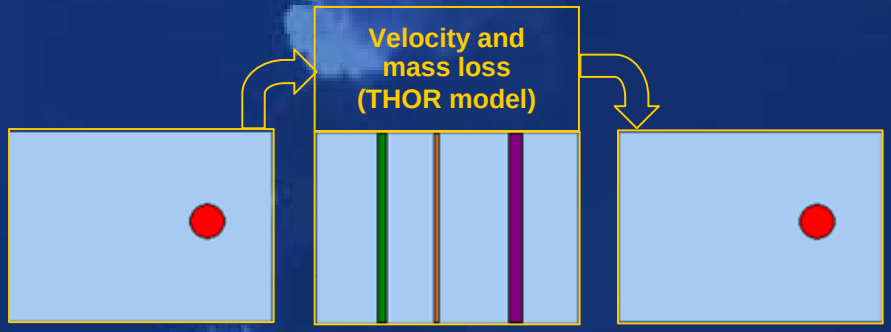
POST-PROCESSING

- **Aim of the paper**
  - This paper details the ongoing work at CEG and SME/CRB to prepare the next version, which should be made available to MSIAC in late 2009, and give details on implementation procedures and coding strategies of interest for potential developers.
    - A companion presentation by P.-F. PERON will describe the work done at MSIAC to implement new models.

Simulation logic

- **TEMPER : as simple as S / M / S**
  - TEMPER decomposes safety problems into the description of a Stimulus / a Mitigation / a Structure. The simulation then runs with one or more Model(s).
- **The simulation logic relies on 2 steps :**

①



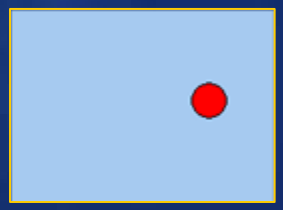
STIMULUS
Spherical fragment

MITIGATION
Spaced plates

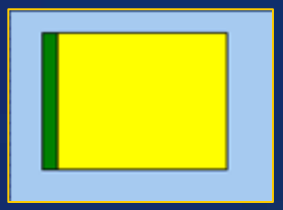
RESIDUAL STIMULUS
Spherical fragment

②

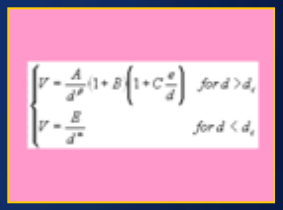
RESIDUAL STIMULUS
Spherical fragment



STRUCTURE
Covered E.M.



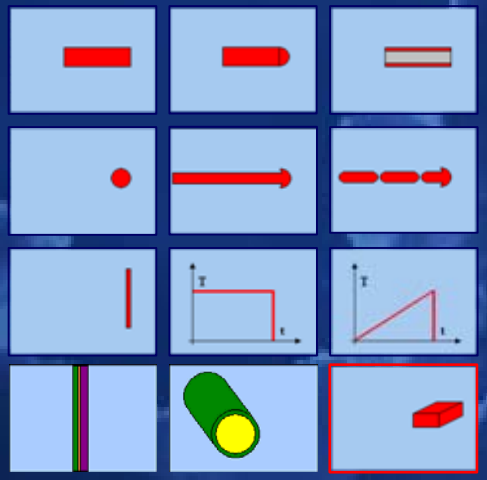
MODEL
V_{lim} Jacobs-Roslund



Implemented models and « objects »

STIMULI

Flat end rod
Round end rod
Flat cookie-cutter
Spherical fragment
Simple shaped charge jet
Real shaped charge jet
Thin plate
Constant Temperature
Rising Temperature
Multilayer Impactor 1D
One on One Warhead
Parallelepiped Fragment

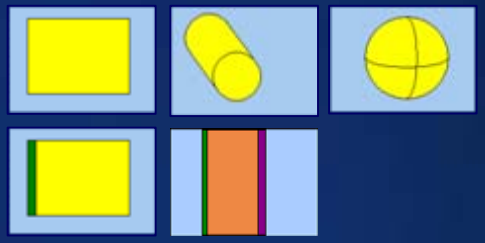


MITIGATIONS

Air **[modified]**
Spaced plates
Single layer

STRUCTURES

Bare plane explosive
Bare cylindrical explosive
Bare spherical explosive
Covered plane explosive
Multilayer Structure 1D



MODELS

V^2d (Held)
u^2d (Held)
Ecrit Walker - Wasley
Ecrit James
Y (Yactor) **[modified]**
Vthreshold (Jacobs - Roslund)
tcook-off (Creighton - Victor)
Ethreshold (Peugeot)
BSDT (Peugeot)
Godlag 1D (Baudin)

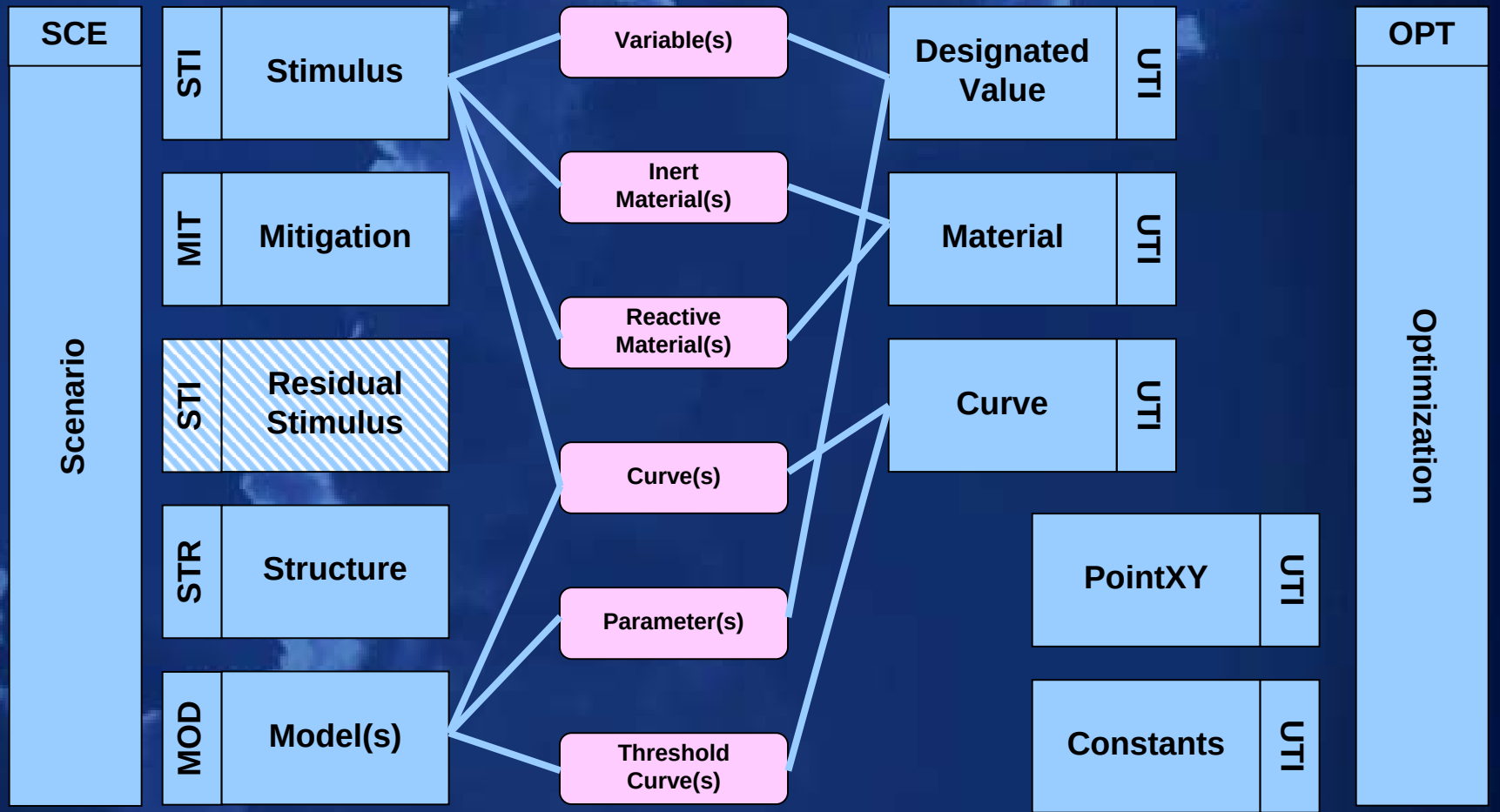
In RED :
New in v2.0

Object-Oriented Programming

- Linear programming
  - Basic variables (Int, Single, double, Char, …) and arrays.
  - Input ⇒ Data processing ⇒ Output.
  - GOTO / GOSUB = almost impossible to debug and maintain.
  - For geeks and computer scientists ! (one-liner contests, vi, ...)
- Structured programming
  - Complex variables (Recordset, List, …).
  - The “main” program calls subroutines when necessary.
  - Easier to debug and partially reusable.
- Object-oriented programming
  - Objects are much more than variables : they have functions and procedures [methods] as well as variables [properties].
  - Program objects mimic real-life things, so their interactions and behaviours are easily understood.
  - Easy to debug and re-use.
- Design patterns in OO programming
  - Design rules to ease the interactions between objects (for instance the Model / View / Controller pattern).

The objects model in TEMPER

(somewhat simplified …)

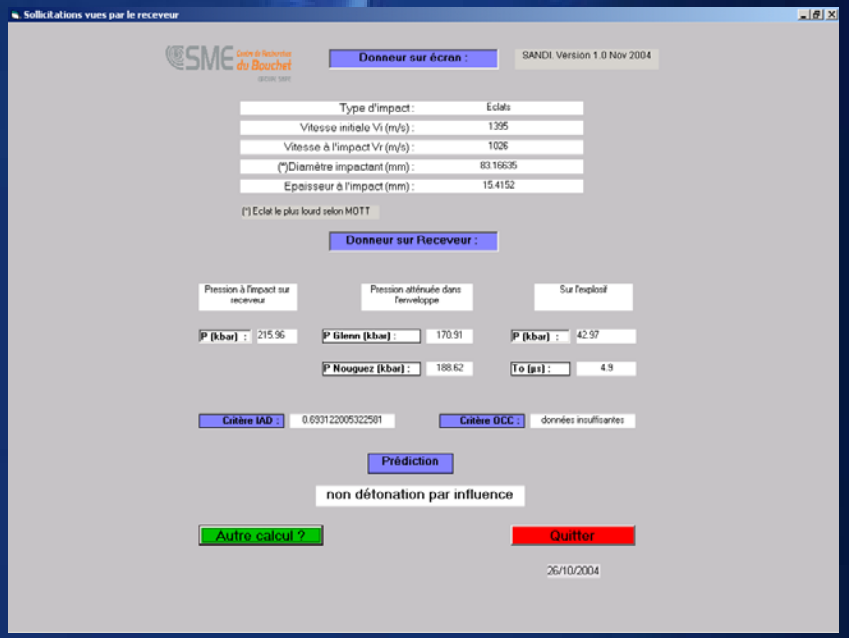


SME/CRB Models

- **SME/CRB : a new developer for TEMPER**
  - **The SME/CRB Research Center, among many other things, is developing ammunition safety models under DGA contracts.**
  - **Albeit only executables were required by DGA, SME/CRB proposed to share its sources with the community and to allow integration of their models in TEMPER.**
  - **Two of these models will be presented : SANDI for Sympathetic Detonation, which will be a second SD model, and INITHER for the 1D modelling of thermal initiation.**
  - **Details of these models will be released in MSIAC reports.**

SANDI : SD model

- **Brief description**
  - SANDI is a stand-alone program implementing the SME/CRB SD approach [1].
- **Example**
  - The test case consists of munitions filled with PBXN-109, with rubber liner and steel case, separated by an air gap and a steel barrier (mitigation).
  - The fragment is a parallelepiped. Its mass is the largest fragment mass according to Mott equations for a 99% confidence level.
  - Shock matching is computed for the residual fragment, and reaction is assessed by either a pressure or a pressure / time criterion.



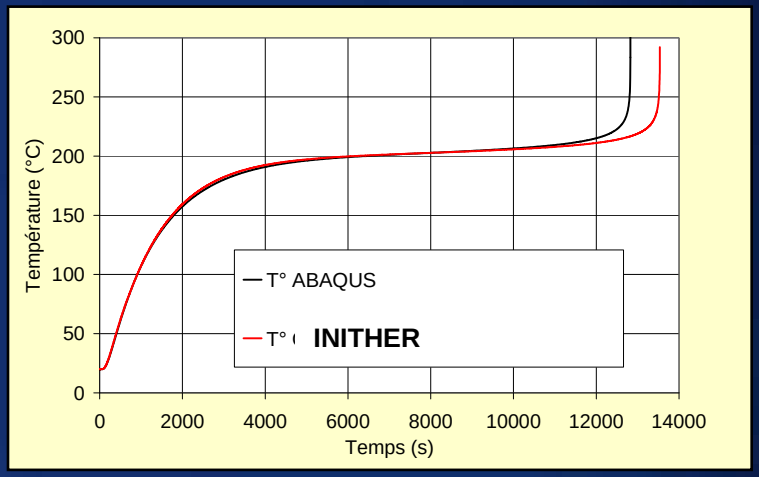
- **Coding strategy for TEMPER**
  - SANDI is a relatively simple model, similar to the MSIAC NDI model. It will be handled in TEMPER as an internal model.
  - The Initial Stimulus will be the “One On One Warhead”, whereas the Residual Stimulus will be a “Parallelepiped Fragment”, previously coded for the MSIAC NDI approach. The “Single Layer” and “Spaced Plates” Mitigations will be adapted to include the new interaction models.

[1] Annereau C., Lécume S., “Modélisation Analytique d’Apparition ou Non d’Une Détonation par Influence”, EUROPYRO 99.

INITHER : 1D thermal model

- **Brief description**
  - INITHER is an 1D finite element thermal code developed by SME for SCO and FCO.
  - INITHER computes thermal exchanges (conduction, convection, radiation) from the outside stimulus to the energetic material through layers of materials in 1D cylindrical geometry.
  - The heat equation is solved in each element by the method of Crank-Nicholson. The reaction kinetics is a three stages Arrhenius.
- **Example**
  - A comparison with ABAQUS is given for the heating of a propellant cylinder having an OD of 50 mm.

ρ= 1820 kg/m3	Cp= 1133 J/(kg.K)	a= 3,627E-07 m2/(s.K)
Z1= 14,31538 s-1	EA1= 12344,867 K	Q1= 57304280 J/m3
Z2= 32,302542 s-1	EA2= 20827,813 K	Q2= -596543500 J/m3
Z3= 32,328255 s-1	EA3= 20731,82 K	Q3= 8486248000 J/m3



- **Coding strategy for TEMPER**
  - INITHER is a more complex model dealing with 1D cylindrical multilayer structures. The model requires time and space discretizations, matrix inversions, etc. To avoid unacceptably long running times, it has been decided to call INITHER as an external executable, using a similar procedure than for GODLAG.

How to include a model ?

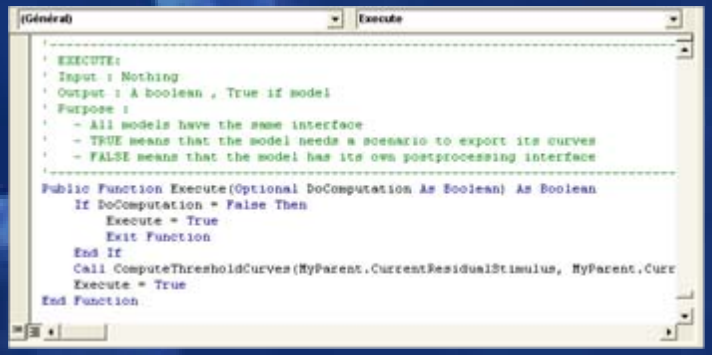
The coding of models is less straightforward than for S/M/S objects, but still relatively simple.

The best way to code a model is to write down equations first, and to wonder how to use existing Stimulus and Structure objects to describe their interactions. It is often possible to re-use part of the code written for similar models (for instance all shock matching for analytical SDT models, a.s.o.).

- **The common structure for Model object classes is as follows:**
  - "Declaration" part [properties, coded once for good]
    - Definition of private "properties" (variables or objects).
  - "Interface" part [access to properties via methods, coded once for good]
    - Allows other objects to access private properties through a controlled channel.
    - The OO way is through "Property Get" and "Property Let/Set" methods.
  - "Common" part [fixed methods, coded once for good]
    - CheckReactiveMaterialParametersOK.
  - "Adaptable" part [methods modified by the developer]
    - Initialize.
    - Execute (modified only for external models : see GODLAG / INITHER for instance).
    - ComputeThresholdCurves.
    - ReactionDiagnosisPoint (sets the Residual Stimulus on the proper plane).
    - ReturnYFromX, ReturnXFromY (if required) : gives the ordinate of a point on a threshold curve with a given abscissa (or the inverse).
    - CompatibiityRules

External models

- **The « Execute » method:**
  - In TEMPER v2.0, we introduced the possibility to call external programs for complex models. It had already been done for GODLAG, a 1D multi layer shock propagation hydrocode. TEMPER doesn't really make distinctions between internal and external models. The embedding of external models is done through the "Execute" procedure of the Model object class.
  - For simple models, the "Execute" procedure is always the same (picture below).
  - For external models this "Execute" procedure becomes more complex, since it may call specific User Interfaces for both the introduction of additional parameters (time limits, sensors, etc.) and the post-processing of results (right).



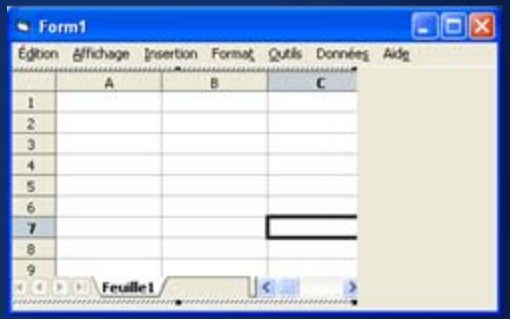
User-oriented developments

- **Bug corrections and Requests for enhancements**
  - The first TEMPER training session has been held in Paris at the beginning of 2009. It was an unique opportunity to discuss with French users. A number of small bugs or requests have been reported and are currently under correction or development.
  - We will do our best to include also in the next release all the bug corrections and (minor !) requests for enhancements that will arise during the 2009 international training sessions (during the present IMEMTS and/or ICT conference).

  "It's impossible to make anything foolproof, because fools are so ingenious …"
- **Better error handling**
  - TEMPER is not error proof and ends sometimes abruptly :
    - Input data check has to be improved.
  - Needs for a systematic Error handling, using Error classes and events.
- **Better unit management**
  - UTI_DesignatedVariable could be updated to really handle conversions.
  - Still a problem for parameter-dependent units ($K = P^n.t$ for instance).
  - Switch to non-dimensional variables as often as possible.
- **Improvement of the online help**
  - Systematic link to MSIAC reports.
  - Inclusion of Flash tutorials (Wink or CamStudio capture softwares)

Pending questions

- **Programming language**
  - VB 6.0 is not supported anymore.
  - "Free" languages may be interesting : Java, Free Pascal, Python.
    - Some of them are multi-platform.
    - Some of them suffer from drawbacks (poor string functions, limited OO, …).
    - Limited OLE support.
    - A huge work is required to recode TEMPER …
  - VB.NET, if not free, has a free Express Edition.
    - Some work is required, since OO is much more drastic than in VB 6.0
    - Express Editions are frequently updated.
- **Embedded Excel workbook : pros and cons**
  - It's nice to have Excel in TEMPER :
    - Post-processing of TEMPER results is immediate.
    - TEMPER benefits from Excel possibilities (multi-sheet, charts, etc.)
  - But the object library in Excel changes frequently and is unstable :
    - Sometimes we loose VBA string functions (documented bug, no solution).
    - TEMPER won't run with Excel 2007 (OLE support ?).
  - Alternatives are :
    - Combined VB MSFlexGrid and MSChart controls.
    - OLE link to an OpenOffice workbook.



Conclusion

- **Also on our roadmap**
  - New projectile (conical tip) and projectile/structure interactions (modified models):
    - MSIAC development : see companion presentation by Pierre-François Péron
  - DDESB blast wave tool.
    - Eric J. Deschambault and Michael M. Swisdak kindly allowed us to introduce the DDESB bare explosive / open air fits in TEMPER.
    - This tool could be used to compute a blast from a « Bare Charge » stimulus and subsequent reaction thresholds, or just to determine blast wave parameters in Air (Mitigation) at different distances.
    - For more complex simulations, the DDESB « Blast Effect Computer » is a valuable and straightforward tool !
- **TEMPER availability**
  - Users and developers are neither within the same location nor within the same organization. In order to provide to the community a common tool that could become a reference in the S3 community, it had been 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, new parameters but also with new ideas/requirements.
    - The developer level, i.e. super users that also use this platform to develop models and share the newly implemented models but not necessarily the parameters coming with.
  - Please contact MSIAC if you are interested to join us !