

Internet based simulations of debris dispersion of Shuttle Launch

Rajkumar Thirumalainambi

SAIC @ NASA Ames Research center, Mail Stop 269-2, Moffett Field, California, 94035

Jorge Bardina

NASA Ames Research center, Mail Stop 269-2, Moffett Field, California, 94035

Copyright © 2004 SAE International

ABSTRACT

The debris dispersion model (which dispersion model?) is so heterogeneous and interrelated with various factors, 3D graphics combined with physical models are useful in understanding the complexity of launch and range operations. Modeling and simulation in this area mainly focuses on orbital dynamics and range safety concepts, including destruct limits, telemetry and tracking, and population risk. Particle explosion modeling is the process of simulating an explosion by breaking the rocket into many pieces. The particles are scattered throughout their motion using the laws of physics eventually coming to rest. The size of the foot print explains the type of explosion and distribution of the particles. The shuttle launch and range operations in this paper are discussed based on the operations of the Kennedy Space Center, Florida, USA. Java 3D graphics provides geometric and visual content with suitable modeling behaviors of Shuttle launches.

INTRODUCTION

Over the past several years there has been a steady increase in space launch operations. NASA is developing operational concepts that will seamlessly integrate air and space launch/reentry operations while ensuring that the highest levels of safety, security and efficiency are maintained. For launches at Kennedy Space center and Cape Canaveral, range safety is maintained from the range operations control center located at Cape Canaveral Air force station [1,12,13,14,15]. Debris dispersion is a key safety consideration in the launch decision process, and understanding the vehicle's design characteristics, atmospheric conditions, and other factors that can affect a hazard impact area is a critical issue for range operators. To evaluate and mitigate risks within the range, range operators use the Common real time debris foot print system. In general, a debris dispersion model computes the velocity of fragments, size distribution, and the impact probabilities and risk associated with debris resulting from a catastrophic failure or intentional

destruction of a vehicle [7, 17]. Presently the eastern range uses the CRTF (Common Real-Time Debris Footprint) [5] program operating inside the RRAT (Range Risk Analysis Tool) to simulate debris dispersion and perform the risk analysis. In the next section, the basic physics of falling debris are discussed.

MECHANICS OF DEBRIS FALL AND DISPERSION

A piece of debris has an initial state vector that is defined by a position and velocity vector. The initial state vector can be perturbed from by an explosion that imparts a velocity and a consequential adjustment to the velocity vector. There is no adjustment to the initial position because the velocity is added impulsively. The dominant parameter in the trajectory computation is the ballistic coefficient [10]; a measure of an object's ability to overcome air resistance. Objects with low weight to drag ratio (low ballistic coefficient) fall much more slowly than objects with a high weight to drag ratio (high ballistic coefficient). Hence, fragments with low ballistic coefficients will tend to travel further given an initial horizontal velocity. In this situation wind becomes a major factor affecting the fall of the debris.

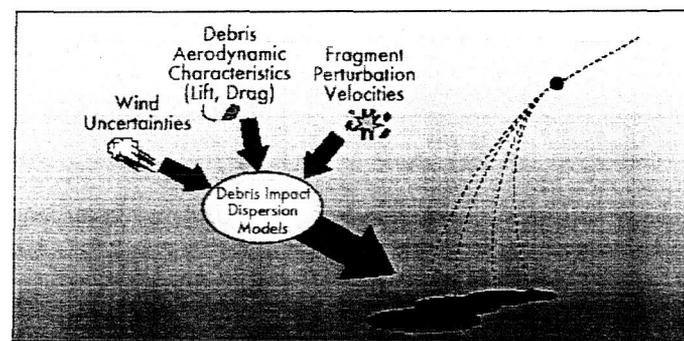


Figure 1 Generic Debris Dispersion Model

The dispersion of debris impact location associated with a given initial state vector is illustrated in figure 1. The primary sources of debris dispersion are ballistic coefficient, wind and velocity perturbation. Wind data is

obtained from Rawinsonde data [8] with upper air characteristics. The following equations provide dispersion of particles w.r.t x,y,z position of translation in 3-D space.

$$X = X_p + V_x \cdot \Delta t + W_x \cdot \Delta t + V_x \cdot C_x \cdot \Delta t$$

$$Y = Y_p + V_y \cdot \Delta t + W_y \cdot \Delta t + V_y \cdot C_y \cdot \Delta t$$

$$Z = Z_p + V_z \cdot \Delta t + W_z \cdot \Delta t + V_z \cdot C_z \cdot \Delta t - 0.5 \cdot 9.81 \cdot \Delta t^2$$

Where p indicates the position where explosion occurs (you state above that the explosion is an "impulsive" process, therefore it doesn't have a "start" or "end"), W indicates the wind local velocity and direction impact on the individual particles (Question: does this include the ballistic coefficient)?, C expresses the coefficient of explosion and explosion type, which is directly related to the dispersion effect. Δt is the time increment of the simulation.. The simulation treats each piece of debris as a unique projectile and uses the above projectile equations to calculate its state at each time step. This

simulation runs until all the particles hit the ground. (Question: do you consider spherical geometric effects?) The air resistance and elasticity effects are also added to particles for a reality effect.. In the model, the launch pad at Kennedy Space Center is considered as point of origin (0,0,0). The reference co-ordinate system X refers to east, Y pointing to the true north and Z is the altitude. All measured units in the model are presented in SI units. With the above basic physical theory and concepts, a JAVA 3D visualization for debris dispersion is developed and explained in following sections.

JAVA 3D MODEL

This application has been developed as an Applet which incorporates physical model in Java which computes rocket characteristics like velocity, altitude, downrange etc. and Java 3D dispersion model [16]. The graphical user interface and output of dispersion model is shown in figure2.

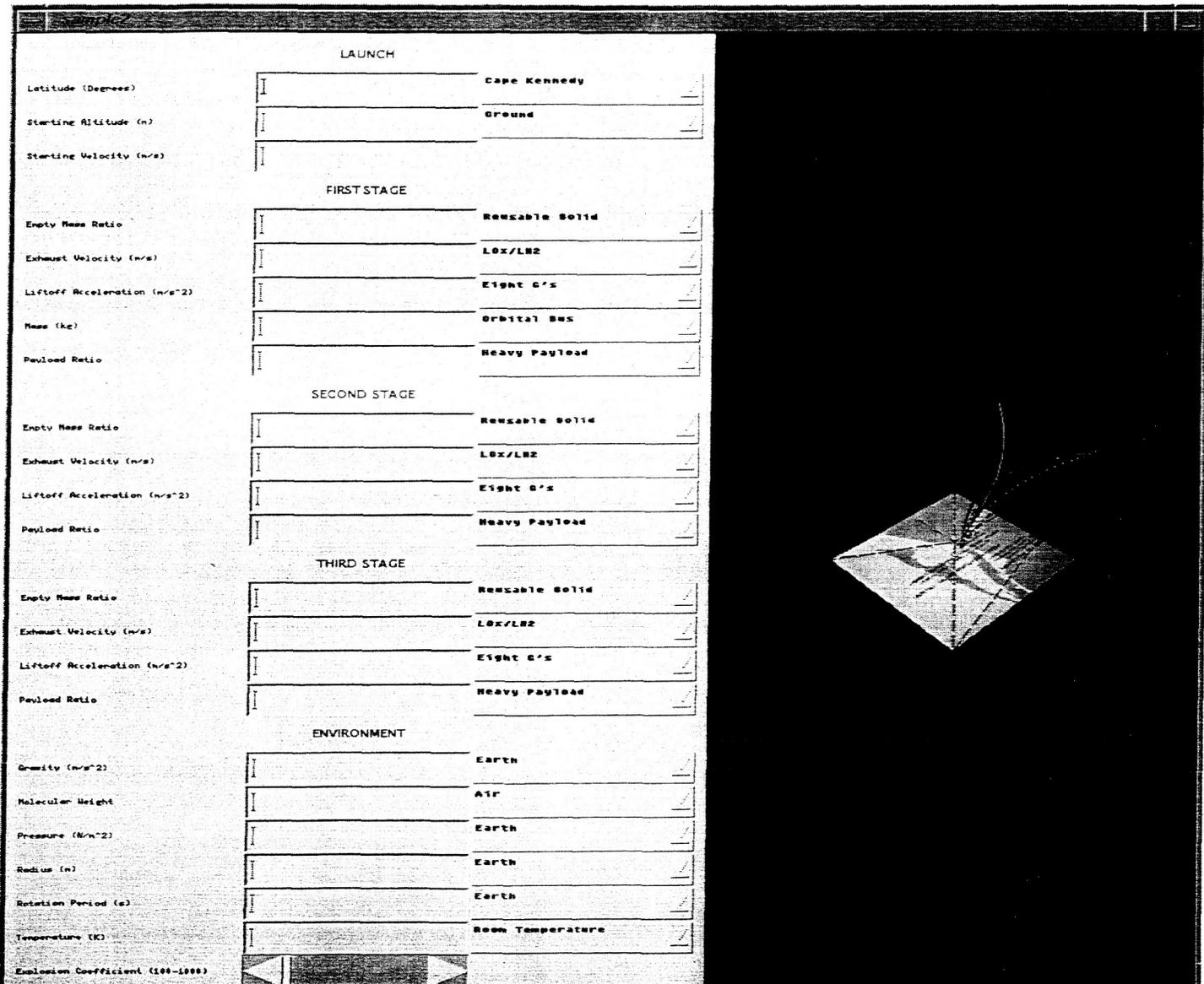


Figure 2. Graphical user interface of debris dispersion model

The left hand side of the the applet accepts launch and rocket characteristics that allow computation of rocket velocity and down range distance. The results are displayed as a 3-D animation showing the rocket's trajectory and destruct lines [2,4]. The user is allowed to specify up to a three stage rocket. There are three stages allowed, but the visualization effect of debris dispersion model applies for the first two stages of launch.

MULTISTAGE TRAJECTORY ANALYSIS

The general characteristics of launch details like latitude of launch pad (degrees), starting altitude (m), starting velocity (m/s) are provided in the input form in figure 2. The specific stage details like empty mass ratio for a given stage, exhaust velocity (m/s), liftoff acceleration (m/s^2), payload ratio and initial mass (kg) are required to compute trajectory details. Airspeed is the speed of the spacecraft relative to the launch site. Airspeed along with the deceleration; which is derived from the density and frontal area of the spacecraft, is used to calculate the air resistance.

$$\text{air resistance} = \text{deceleration} * \text{airspeed}^2$$

When a spacecraft reaches low earth orbit, it is generally at an altitude of one hundred and forty kilometers. A high starting altitude is desirable because the air pressure and density decrease as the altitude increases. A lower air density means the aerodynamic drag will be lower and a lower air pressure means the exhaust velocity will be higher. Altitude is used to calculate air density and orbital velocity.

$$\text{air density} = \text{sea level density} * \exp(- \text{altitude} / \text{column height})$$

$$\text{orbital} = \text{ground orbital} * \text{sqrt}(\text{radius} / (\text{radius} + \text{altitude}))$$

Column height is defined as the average column height of the atmosphere; the height rise after which the pressure falls by a factor of e. This is calculated using gravity, molecular weight and temperature. This is then used to derive the air density as the spacecraft rises. The Radius refers to earth's radius.

$$\text{Column height} = 8314 (J * K / \text{kmol}) * \text{temperature} / \text{gravity} / \text{molecular weight}$$

$$\text{Air density} = \text{sea level density} * \exp(- \text{altitude} / \text{column height})$$

Downrange is the horizontal distance traveled by the spacecraft. When a spacecraft reaches low earth orbit, it is generally downrange about nine hundred kilometers, however it is dependant on several launch conditions. The downrange when reaching orbit will decrease as the starting velocity increases. The downrange when reaching orbit will increase as the acceleration decreases. For example, a low acceleration space plane will be far downrange when reaching orbit, while a high thrust rocket will be relatively close to the launch position.

The velocity and downrange distance are computed using above equations, which form the initial velocity, altitude and down range distance from launch pad when explosion occurs. In this model, the number of particles is assumed as 500 random size pieces. The user can alter any assumptions of payload, vehicle types and launch pad scenarios in Kennedy Space Center.

VISUALIZATION

The scene graph consists of superstructure components, a *VirtualUniverse* object and a *Locale* object, and a set of branch graphs which is displayed via 3D canvas [4,6]. The simulation is done via Java 3D behaviors. There are three behaviors which correspond to orbit, explosion and dispersion of particles. Behaviors allow specifying an action based on a particular set of events called the *WakeupCondition*. A *WakeupCondition* is a combination of another Java 3D object called a *WakeupCriterion*, which covers a wide range of activities like keyboard stroke, frames, waiting, etc. The *spaceHit* behavior enacts a key board strike, which removes the shape of the rocket from the scene graph, and adds a '*createexplosiongraph*' branch graph which displays a representation of an explosion. The next key stroke (what was the first key stroke?) invokes the *explosion* behavior, which is a specialized interpolator. This interpolator describes the dispersion of the particles from the point of explosion and scattering around the path of dispersion to the ground continuously. The transition from the starting to ending value is interpolated continuously from the value 0 – 1.0. The *Alpha* class is a simple object that help Java 3D worlds come alive. *Alpha* object generates a single ramp between 0 and 1.0. There is a random number generator allocating which trajectory rocket should be launched. If rocket entered into an abnormal trajectory or crossed destruct limit lines, *spaceHit* behavior is invoked to kill the rocket. The *VirtualUniverse* for debris dispersion model is shown in figure 3.

If the rocket orbits in a perfect trajectory and corresponding *Alpha* reaches 1.0, the satellite orbit is activated in another universe (??). The trajectories are constructed by *Bezier* curves [11]. The origin, end point and another two points which represent between the start and end points are called control points. The curves are constructed as a sequence of cubic segments, rather than linear ones. The entire curve is contained in the quadrilateral whose corners are the four given points (their convex hull). Constructing these curves is very efficient due to the employment of a simple recursive algorithm that reduces the computation of points to a simple "division-by-two" operation. Seven trajectories are constructed and dotted lines represent destruct limit lines. The bottom texture of the box has the appearance of Kennedy Space Center with launch pad locations in figure 4.

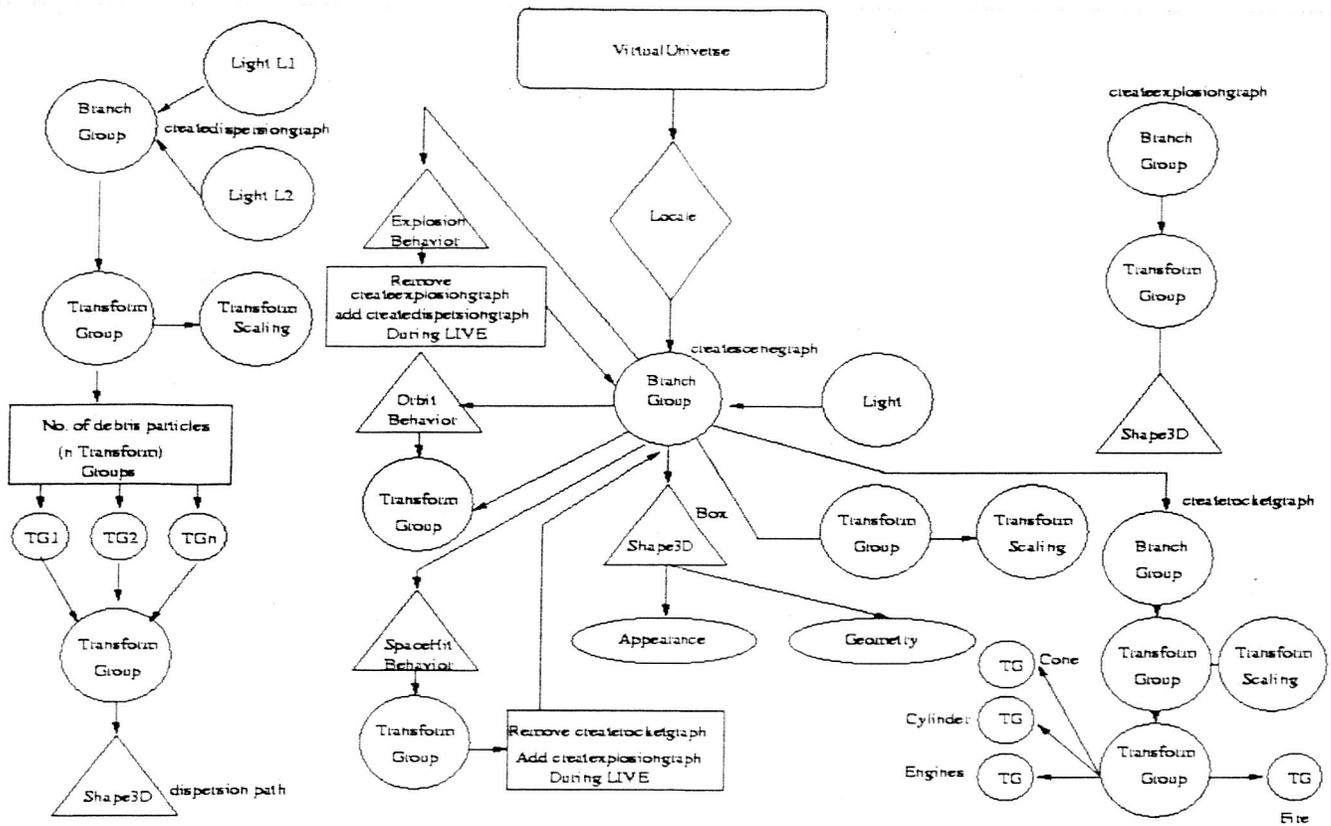


Figure 3 Scene graph for Shuttle launch and debris dispersion model

LAUNCH	
Latitude (Degree)	CAPE KENNEDY
Starting Altitude (m)	GROUND
Starting Velocity (m/s)	
FIRST STAGE	
Empty Mass Ratio	REUSABLE BOTTLE
Exhaust Velocity (m/s)	LOX/LR2
Lift-off Acceleration (m/s ²)	EIGHT G'S
Mass (kg)	ORBITAL BUS
Payload Ratio	HEAVY PAYLOAD
SECOND STAGE	
Empty Mass Ratio	REUSABLE BOTTLE
Exhaust Velocity (m/s)	LOX/LR2
Lift-off Acceleration (m/s ²)	EIGHT G'S
Payload Ratio	HEAVY PAYLOAD
THIRD STAGE	
Empty Mass Ratio	REUSABLE BOTTLE
Exhaust Velocity (m/s)	LOX/LR2
Lift-off Acceleration (m/s ²)	EIGHT G'S
Payload Ratio	HEAVY PAYLOAD
ENVIRONMENT	
Gravity (m/s ²)	EARTH
Molecular Weight	AIR
Pressure (N/m ²)	EARTH
Radius (m)	EARTH
Rotation Period (s)	EARTH
Temperature (K)	ROOM TEMPERATURE
Explosion Coefficient (100-1000)	

Figure 4 Debris dispersion model visualization

The debris dispersion is displayed in figure 4 showing that all debris particles fall into the elliptical impact zone (orange cluster of particles in right hand side over Florida and Atlantic Ocean). The debris dispersion model obtains Rawinsonde data [8] from an online weather server and incorporates wind effects into debris dispersion model computations shown above. The three dimensional visualization provides the debris scattering impact zone the potentially affected areas. Based on debris impact, human health risk assessment and expected casualty can be derived [3,9,19].

CONCLUSION

Earlier debris dispersion model developed are based on two-level full factorial analysis on the CRTF [5] debris dispersion using experimental design techniques [18]. From the above model, the results show that three factors have statistical significance; ballistic coefficient of debris fragments (C), failure altitude (F) and wind speed and direction (W). In the model presented, the Java 3D model shows a simple orbital and debris dispersion model based on physics with necessary parameters and their combined interactions in each fragments. The model will be used by Range Safety Officers as a simulation tool during decision making process.. Future research will focus on an updated dispersion model combined with all available parameters using Java 3D. Future model make a decision to destroy Shuttle when it violates flight launch rules with built in intelligent systems. Developing the application in Java 3D aids in its distribution and usefulness across all NASA centers.

The validation of models is based on telemetry and radar data to monitor orbital dynamics and any debris dispersion. Outside the Shuttle program, it can be used for missile tracking or any trajectory analysis. The visualizations are self-explanatory and can be interpreted easily by user. This visualization tool can be analyzed collaboratively using web. The applications of this tool can be used for studying any catastrophe in launch and range operations in lunar and mars surface. This model can also be integrated into onboard systems for telemetry tracking and onboard destruct mechanisms based on flight rules.

ACKNOWLEDGMENTS

We would like to acknowledge useful comments, suggestions and discussions from and . The authors also wish to thank Dr. Michael Shafto and Dr. Butler Hines for their financial support through the CICT program at NASA Ames Research Center.

REFERENCES

- Bardina, J. and T. Rajkumar. 2003. "Intelligent launch and range operations Virtual test bed (ILRO-VTB)" In *Proceedings of SPIE Enabling technologies for simulation science*, vol. 5091 141-148 Florida, USA.
- Baskett, R and J. Pace. 1995. *ARAC Dispersion modeling for January-March 1995 Vandenberg AFB Launches* Lawrence Livermore National Laboratory. USA.
- Bennett, R. R. and McDonald, A. J. 1999. "Local environmental and toxicity issues for rocket launching and testing". JANNAF 28th propellant development and characterization subcommittee and 17th safety and environmental protection subcommittee joint meeting. 153-173., USA.
- Brown, K. and D. Petersen. 1999. *Ready-to-Run Java 3D*. Wiley Computer Publishing. USA.
- Carbon, S. 2001. "Common real-time debris footprint", ACTA Report No01-451/23.1-01, September 2001.
- Davidson, A. *Java Games Programming Techniques*. <http://fivedots.coe.psu.ac.th/~ad/jg/>.
- FAA. 1999. *45th Space wing/Patrick air force base launch site safety assessment*. Prepared under contract no. DTFA01-95-D-03009. Licensing and Safety division, Washington D.C. USA.
- Florida-State-University, 1998. <http://www.met.fsu.edu/cudos/rawins/>
- Hudson, J.M; A.M. See; and L. L. Philipson. 1999. "Launch Area Toxic Risk Analysis (LATRA): Risk management computer Program". JANNAF 28th propellant development and characterization subcommittee and 17th safety and environmental protection subcommittee joint meeting, 183-191., USA.
- Jensen, J; G. Townsend; J. Kork; and D. Kraft. 1962. *Design guide to Orbital Flight*. McGraw Hill Co. USA.
- Lengyel, E. 2004. *Mathematics for 3D game programming and computer graphics*. Charles River Media Inc. Second Edition. USA.
- NASA. 1988. *Range safety 1102*. Mission Directorate. Flight analysis branch. USA.
- NRC. 1998. *Assessment of Exposure-Response functions for Rocket-Emission Toxicants*. National Academy Press. Washington D.C. USA.
- NRC. 2000. *Streamlining space launch range safety*. National Academy Press. Washington D.C. USA.
- Range commanders council standard 321-02, 2002. "Common risk criteria for National Test Ranges", Secretariat of the RCC US Army White sands Missile Range, NM 88002-5110.
- Selman, D. 2002. *Java 3D Programming*. Manning Publications Co. USA.
- Turco, M, Krell B, Philipson, L and Hudson, J. 1995. "Launch Vehicle Failure analysis and database management system" Report No 95-314/72-10, Prepared for the department of the Air Force, 30, SW, Vandenberg AFB and 45 SW, Patrick AFB, ACTA Inc, California.
- VanSuetendael, R and Rekab. K. 2004. "Evaluating a complex debris dispersion model using experimental design techniques" In *Journal of aerospace computing, information and communication* Vol 1 pp 100-107, February 2004.

19. Yassi, A. 1998. Basic Environmental Health. World Health Organization.