1-1-1993

DIS/J-MASS Interoperability

Rue Hestand

Brian Goldiez

INSTITUTE FOR SIMULATION AND TRAINING

June 16, 1993

DIS/J-MASS Interoperability

Rue Hestand • Brian Goldiez

IST

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

IST-TR-93-19

INSTITUTE FOR SIMULATION AND TRAINING

DIS/J-MASS Interoperability

Rue Hestand • Brian Goldiez

June 16, 1993
IST-TR-93-19

Institute for Simulation and Training • 12424 Research Parkway, Suite 300 • Orlando, Florida 32826
University of Central Florida • Division of Sponsored Research

There are two simulations which are to be interfaced to demonstrate interoperability. They are DIS (Distributed Interactive Simulations) and J-MASS (Joint Modeling and Simulation System).

DIS is a simulation which operates in "real time." Each computer in DIS runs its own simulation and controls one or more "entities", which are the main elements of the simulation. Entities can include tanks, submarines, carriers, fighter aircraft, missiles, etc. Individual nodes operate autonomously and pass information. The computers communicate with one another by issuing PDU's (Protocol Data Units). All information necessary to run the distributed simulation is contained in these PDU's which are passed between nodes on the DIS network. Examples of PDU's which are or will be included are Entity State, Fire, Detonation, Service Request, Resupply Offer, Resupply Received, Resupply Cancel, Repair Complete, Repair Response, Collision, etc. There are 27 PDU's defined (March 22, 1993) however all are not implemented at this time.

J-MASS does not operate in real time. Another major difference with DIS is J-MASS has one "master computer" and many "slaves." The master computer specifies tasks which the slaves execute as well as controls the simulated environment. This means there is one main computer running the simulations and the other computers act as support. The J-MASS equivalent to a DIS entity is a "player." A player is defined as anything which, at some point in the simulation, functions autonomously in space. The two players which are currently modeled in J-MASS are aircraft and missiles.

The purpose of this effort is to build an interface between these two simulations. The major assumption being used is that J-MASS will run at real time or faster. This assumption being true there are still a multitude of problems which may occur. The following report identifies issues of DIS/J-MASS interoperability that were unclear or that may be sources of error. The report includes suggested solutions to the problems when applicable, as well as information intended to clear up obscure or missing areas. Three of

the major areas IST looked into dealt with the synchronization of the teams, the coordinate systems used by J-MASS and DIS, and the earth models used by J-MASS and DIS.

Appendix A following the report is an analysis of the error J-MASS incurs when calculating distance on the surface of a spherical earth in the region of Fort Hunter Ligget versus the distance using an oblate ellipsoid earth (WGS84 model that DIS uses). The distances were referenced by two points in a latitude, longitude coordinate system (geodetic with the elevation being zero).

References:

Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993.

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

J-MASS Coordinate Systems Standards. Version 1.1 (J-MASS-CSS-1.0) 16, August 1991.

J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

Lin, Kuo-Chi and Ng, Huat Keng, "Interconversions Between Different Coordinate Systems", Institute for Simulation and Training.

Military Handbook on Datum, Projections, Grids and Common Coordinate Systems, Transformation of DoD (MIL-HDBK-600008). May, 1991.

Rapp, Richard H., "Geometric Geodesy, Part 1." Ohio State University - Department of Geodetic Science and Surveying. April, 1991.

Science Applications International Corp., Aeronautical Systems Operation-Dayton, OH Memorandum, dated 2 June 1993, Log WBM:062, from Barry Michael to Onda Simmons.

Software Structural Model Design Methodology for the Modeling Library Components of the J-MASS program. Version 1.1. 1, April 1992.

Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

System/Segment Design Document for the J-MASS Program. Version 0.0, (J-MASS-SSDD-0.0). 18, May 1992.

System/Segment Specifications for the J-MASS Program. Version 1.1 (J-MASS-SSS-1.1). 7, September 1992.

1. *Issue:* The handling of Destroy Events between J-MASS players and non-DIS players (including munitions) must be examined because they are handled differently.

Reference: Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993.

Disposition: J-MASS handles all Destroy Events (Collisions, Detonations, etc.) the same way, by removing the players from the simulation. DIS can only detect the absence of ES PDU's, therefore DIS will see a player for several time units after it's already been removed from J-MASS. (ES PDU's are sent when certain space and time thresholds have been exceeded. DIS will think the entity is stationary until the time threshold is exceeded and it expects another ES PDU and there is no guidance in DIS on the handling of ES PDUs which exceed time thresholds). To avoid this problem make the removing of players an "event", which, when sent through the event processor, will return a ES PDU with spatial coordinates corresponding to the center of the earth. This will effectively remove the players from the DIS side.

Collision handling can be done in a variety of ways. The two extremes are ignoring collisions or using a pure Newtonian physics model. All other methods fall somewhere between those two extremes. Currently J-MASS and DIS are closer to ignoring collisions than using the Newtonian model. DIS is closer to the Newtonian model than J-MASS. J-MASS and DIS need to brought to the same modeling level, and they both need to come closer to the Newtonian model. DIS is already moving in this direction. IST is in the process of determining how much added collision handling should be done in order to maintain a good balance between processing speed (must be fast enough for real time) and how realistic the simulation is. A more realistic simulation implies a more detailed level of modeling and thus more processing must be done.

2. *Issue:* Handling of collisions between J-MASS players and DIS entities.

Reference: Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993.

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Disposition: When J-MASS and DIS detects a collision between a player and an entity, both J-MASS and DIS should issue a Collision PDU. J-MASS should remove the players as it usually does and issue an ES PDU moving both the entities to the center of the earth, where they'll be out of the way for all practical purposes. Collisions will not occur for the initial DIS/J-MASS demonstration.

3. *Issue:* Response when two DIS entities collide.

References: Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: When two DIS entities collide, both issue a collision PDU (or one issues a Detonate PDU if one is a missile). The entities assess the damage based on information within the PDUs, and DIS entities report the extent of the damage via the ES PDU. If one or both entities are destroyed a Destroy Event needs to be generated for J-MASS and it will remove the DIS player or players from its side of the simulation. DIS should move the destroyed entities to the center of the earth. No J-MASS event will be generated in the case where the entities are damaged. This won't matter if DIS either takes care of a partially working system or just allows the system to be inoperative and not use it.

Example: If a tank is damaged it may not move as fast. J-MASS will still see the ES PDU's. It doesn't matter if the tank is damaged and moving at its present full speed which may be half the original speed, or if the tank is undamaged and moving at half speed.

If the radio or radar is damaged DIS can just not use the system if it's too difficult to model the effects of such damage. Note: DIS presently does not handle any RF events. For a part which is unusable (radio, radar, etc.) the simulation can remove it from the ES PDU. Currently, only articulated parts and attached parts and stores for visual appearance are in the ES PDU's. In the future internal fixed parts and stores and their position in Entity Coordinates should be added to the ES PDU.

4. *Issue:* It was unclear how the synchronization process between teams was managed.

References: Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993.

J-MASS Ada Simulation Runtime Agent and Interim SSE Review 25-25, January 1993.

Disposition: The synchronization of the teams in the simulation is split across the Simulation Executive and the individual teams. Each teams' synchronizer calculates its Minimum Response Time (MRT) and the Next Event Time (NET).

NET - This is the time difference between current simulation time and the time of the next event in a team's event queue. It's possible to have a NET larger than the time of the actual next event.

Example: NET = 8 for Team 1 (this could be the time the player will update its radar). At time 4 a player in Team 2 moves closer to Team 1. This causes a player in Team 1 to fire at Team 2 at time 5. Thus the next actual event was at time 5 even though the NET = 8.

The NET isn't necessarily the time of the next event but the time of the next event in the event queue.

MRT - This is the time a team may process without generating events which another team will need to know about. This must be guaranteed (refer to Issue 5 on the guaranteeing of the MRT). In the example above the event which will take place at time 5 for Team 1 is an event that another team needs to know about. Therefore Team 1's MRT = 5.

When the team synchronizer finishes those calculations it sends them to the executive synchronizer. The executive synchronizer then calculates the Clock increment and the Update Cycle.

Clock Increment - The amount of time to increment the system clock before an event and event processing occurs. The smallest NET out of all the teams will be the Clock Increment.

Update Cycle - The amount of system clock time a team may independently process its events. This is the smallest MRT of all the Teams.

The executive synchronizer waits until the clock time is equal to the current time plus the clock increment and then sends an Advance message to all Teams. This "new time" puts the simulation back on Real Time for the event about to take place. This is necessary to insure interoperability with the Real Time DIS.

The teams then process all the events in their queue from the "new time" to the "new time" plus the update cycle. When that time difference has elapsed, we are at the time where a team has an event others need to know about (the old MRT). Then each Teams' synchronizer calculates a new MRT and NET and the process repeats. Note: Events in the event queue are prioritized based on first time and then Communication, Update Player, Spatial, Data Journal and then Destroy Events.

5. *Issue:* The MRT is supposed to be guaranted, but it is very difficult to actually guarentee the MRT. However, there may be a way to obtain the MRT without a great deal of computation.

Reference: Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Disposition: Referring to the example in Issue 4 under the NET, we saw that the MRT could be less than the NET. This occurs when Team 2 does something which causes a MRT event (an event which 1 or more other Teams need to know about) to be generated by Team 1 at a time before the time of the next event in the event queue. This means that to guarantee the MRT all events would have to be processed internally up to the time of greatest NET of all Teams. This is because the largest NET of all the Teams must be larger than the MRT for the Team whose MRT is less than its NET. Events which are generated due to those processed events must also be kept track of in case one of them occurs *before* the Team's NET *and* is an event that other Teams need to know about. In that case, the time of that event (or if several events were generated that meet those requirements, the earliest time of the events) will be the MRT. This requires a great deal of extra processing and may be a serious problem. The Team Synchronizer probably isn't equipped at this time to handle that situation.

Since any event that causes a MRT event should also be a MRT event, it's not necessary to do all the extra processing. The smallest MRT, which is our only concern (the Update Cycle), must be in the event queue. In other words, the case where the smallest MRT of all the Teams, say Team 1, occurs before the NET of that Team (Team 1) can *not*

happen. Since the smallest MRT event must be in the event queue already, it's not necessary to process the events to see what events may be generated and placed on the event queue.

6. *Issue:* During synchronization a MRT and NET must be calculated. This implies that several events in the future are known, which in turn implies the scenario is "canned" or deterministic. This may cause problems with DIS since it is designed to be non deterministic in nature.

References: Fax from Amherst Systems : SRA with DIS Manager, Concept of Operations. 30, March 1993

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Disposition: The initial MRT and NET are set to ∞ for the DIS Team. This is done because the DIS Team has no update events which would be placed on the event queue immediately and because we want J-MASS controlling the simulation, since in the near term it will handle most of the actual computations. The DIS events that occur will usually be in response to some other event, which means DIS doesn't really need to be deterministic. However, the Team Synchronizer in some respects must be. In the near term the DIS Team will be used primarily for flying targets and these targets will react to various events in a certain manner. For example, if the DIS entity is being attacked it goes into an evasive maneuver mode. The DIS Manager, which is seen as a Team, will have a fixed MRT and NET. This will be based on empirical results to allow timely response to DIS PDU's without excessive overhead.

7. *Issue:* During synchronization the Executive Synchronizer uses the Teams' MRT and NET to calculate the Update Cycle and the Clock Increment. This calculation takes some

time as will as the network time involved with all the Teams sending information to the Executive Synchronizer plus the time for the Executive Synchronizer to send information back. This time is not taken into consideration. This can cause problems when the processing time is very close to the Update Cycle.

Reference: J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Disposition: When processing time equals the Update Cycle the simulation will no longer be Real Time. To avoid this problem subtract an estimated time taken by the network traffic and calculation of Update Cycle and Clock Increment from either the Clock Increment or from the Update Cycle. It's better to overestimate the time taken by the network traffic and calculation of Update Cycle and Clock Increment than to underestimate it, since the package must maintain real time for DIS interoperability.

8. *Issue:* The composition and description of J-MASS players was unclear.

Reference: J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

Disposition: J-MASS is built around objects called players. A player is any entity which, in some point of the simulation behaves autonomously. Examples of autonomous behavior: Missile using its tracking system to destroy another player; Aircraft flying evasive maneuvers because its being fired upon; etc. The behavior isn't truly autonomous because it's still being controlled by the computer, but the flight path is no longer predetermined since it depends on the events generated by another player.

Each player is composed of a platform object, control object, and a subsystem assembly object. In general the subsystem assembly object is composed of other subsystem assembly objects and component objects. The component objects aren't composed of any other objects and represent abstract or physical objects. An example of a subsystem assembly is a radar with the radiator, collector, signal generator, etc. being component objects of the radar.

9. *Issue:* The management of intra-player communication within J-MASS and DIS was unclear.

References: J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: Intra-player communication is communication between the player and its various subsystem assemblies. An example is an aircraft sending a message to its missile to deploy and target an airborne interceptor, or sending a message to its radar to switch from search mode to tracking mode. Currently DIS performs no intra-player communication, but for the current design scope that won't be necessary. (Note: The current design calls for a DIS aircraft, a J-MASS aircraft, and possibly a J-MASS SAM. The DIS aircraft will not be firing or modeling any RF communications. The J-MASS players will attempt to destroy the DIS aircraft.) J-MASS controls all its subsystems through intra-player communication.

10. *Issue:* If the maximum time for the simulation is exceeded, then the Executive Synchronizer sends a Terminate message to all of the Teams. Then all the files are closed and the processes stop. What is the maximum time, and how is it determined?

Reference: J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Disposition: Before the program is run a maximum simulation time is defined. This corresponds to the duration of the battle. This is particularly useful, for example, when one wants to analyze the first five minutes of the simulation. Currently this is the only way the simulation will terminate. A possible default method of determining when the simulation should end that may be useful is to terminate when all the players or entities on one side have been destroyed. For one-on-one engagements, which is all the simulation will be used for in the short term, this method of termination will be much more convenient.

11. *Issue:* The way J-MASS and DIS handled the environment was unclear.

Reference: J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.
Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: DIS currently does no centralized environment handling. The only DIS entities which will interact with J-MASS players are airborne, and DIS does not control the atmospheric environment for DIS entities.. This is because the DIS Team only handles ES PDU's (which are converted to Spatial Events) and Collision and Detonation PDU's (which are converted to Destroy Events), and the atmosphere doesn't have the ability to alter the course of an entity (assuming it's not turbulent) or effect the Destroy Events. On land a course of due East (the "y" direction), may also reflect changes in the distance from the center of the earth (the "z" direction) if the terrain isn't flat or if the earth is modeled as an ellipsoid. Thus, environment handling must be employed in order to model land based

entities, or assume a flat terrain which stays flat during the simulation. Since DIS handles no RF communications, the characteristics of the atmosphere aren't crucial.

J-MASS models both the atmosphere and the terrain for electromagnetic (EM) effects and aerodynamic effects on the players. Some of the EM effects considered are the effects of the atmosphere and terrain/sea on direct-path propagation, specular and diffuse forward-scatter (multi-path) reflections, and back scatter.

DIS entities model what they need from the environment individually as if operating in free space. In the future an effort should be made to either allow J-MASS to handle environmental effects of DIS entities or give DIS the capability of doing so. A possible method of insuring that all entities and players are modeling the same environment is to have one model that both DIS and J-MASS can access. Something similar is already being done with the spatial record or State Model. Both DIS and J-MASS use the State Model for spatial information about the players and entities. In DIS, each running simulation would have to be able to access the model. Until DIS can model RF communications or terrain based entities, however, the environment won't be a major concern.

12. *Issue:* The method of identifying other entities or players by DIS and J-MASS.

Reference: J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: DIS currently has no means other than visual to detect other players and entities. It keeps track of the players/entities locations by the ES PDU's. J-MASS has a

much more sophisticated method involving RF communications with radar. The radar has three modes of operation: Search, Acquisition, and Single Target Tracking Modes. In the tracking mode, the modeling becomes much more detailed. J-MASS currently has no visual inputs and DIS currently has no radar input. In the near term this won't cause any problems because DIS will most likely be controlling only target aircraft, and J-MASS will be handling the rest. Our near term goal is to have a DIS aircraft, a J-MASS aircraft with missiles, and a SAM system interact. The DIS aircraft won't do any firing, so J-MASS will perform nearly all the modeling. In the future, additional features will need to be inserted in DIS and J-MASS to cover all model types from both environments.

13. *Issue:* DIS and J-MASS use different spatial management coordinate systems. What are these coordinate systems, and will there be problems with interoperability?

Reference: Lin, Kuo-Chi and Ng, Huat Keng, "Interconversions Between Different Coordinate Systems", Institute for Simulation and Training.

J-MASS Ada Simulation Runtime Agent and Interim SSE Review. 25-25, January 1993.

J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: J-MASS uses many coordinate systems (discussed in Issue 14), but here three major coordinate systems used by J-MASS will be discussed. These are the earth centered, earth Greenwich, and earth surface base coordinate systems. The earth surface base coordinate system of J-MASS is topocentric. In the topocentric coordinate system the origin of the system is on a selected point on the earth's surface and aligned at the selected

point with East, North, and Up being the 3 axes. DIS and J-MASS share the earth Greenwich coordinate system which has its origin at the centroid of the earth. The x axis passes through the Prime Meridian at the equator, the y axis passes through 90° East longitude at the Equator, and the z axis passes through the North Pole. The earth centered coordinate system is very similar to the earth Greenwich coordinate system. The only difference is the earth centered frame doesn't rotate with the earth as the earth Greenwich does. It is fixed with respect to the stars. As far as interoperability is concerned there should be no problem. (Note: DIS uses a different earth model than J-MASS. This problem is discussed in Issue 15.) Algorithms for transformations between coordinate systems exist (see references) so it won't be difficult to transform a spatial point to any coordinate system necessary.

14. *Issue:* J-MASS has a very computationally efficient coordinate system structure. It uses many coordinate systems set up in tree, which it can traverse easily. This enables the programmers to place a coordinate system wherever they want. For dynamic anlysis and many other types of analysis this greatly simplifies the mathmatics of the problem. The method of traversing this coordinate system tree structure, and the coordinate systems J-MASS uses was unclear.

Reference: J-MASS Software Design Document. Version 1.1 (J-MASS-SDD-1.1), 25, March 1992.

J-MASS Coordinate Systems Standards. Version 1.1 (J-MASS-CSS-1.0) 16, August 1991

Disposition: J-MASS uses a tree structure to represent the coordinate systems. The tree has a base coordinate system, and all other coordinate systems are under the base. The coordinate system directly below the base is called the child of the base, and the base coordinate system is called the parent of the child. All the coordinate transformations are

done by traversing this tree. When you have reached the coordinate system within the tree that you want the vector transformed to, you have finished. J-MASS translates the old position vector into the new frame by translation if necessary and rotation using the Rotation_Package (ROT). Then the new position vector becomes the rotated position vector added to the parents record. This rotation package provides a general utility to generate 3-D rotation matrices that can be multiplied together to generate new matrices and/or used to convert vectors from one coordinate system to another if the two coordinate systems have a common origin and are rotated with respect to each other.

The parent/child relationship means that the location, orientation, and motion of each coordinate system must be described in terms of the coordinate system above it on the tree (its parent). Each coordinate system keeps track of where it is, how it is oriented, and how it's moving with respect to its parent coordinate system. Each object computes its motion and position relative to its parent systems and stores it as data to be used for the transformations. We don't know exactly how the vectors are rotated because we don't have any code for the ROT function. Note: J-MASS translates all vectors including linear velocity and acceleration vectors, rotational velocity and acceleration vectors, constant magnitude vectors, orientation vectors, etc.

At the top of the tree is the inertial coordinate system, CS_I. This system is usually approximated by another coordinate system which would normally be a child or grandchild, etc., of CS_I. The earth centered coordinate system, CS_EC, is the child of CS_I. CS_EC has its origin at the center of the earth (as described in Issue 13). It is fixed with respect to the stars. The earth Greenwich coordinate system also described in Issue 13, CS_EG, is below CS_EC. All the effects of Earth's rotation are included in the relationship between CS_EG and CS_EC.

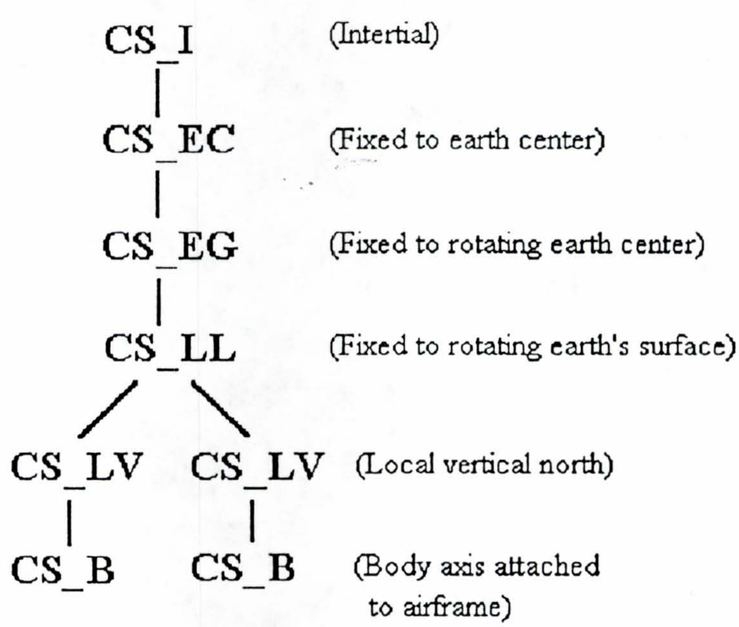
The child of CS_EG is CS_LL (latitude, longitude). This coordinate system is the earth surface coordinate system. The origin is fixed to the earth at some location given by its longitude and latitude. All shape effects are encapsulated in the CS_LL to CS_EG

transformations. In J-MASS the inertial system is currently being approximated as CS_LL. The system should be fixed to a point on the surface of the earth from which the scenario will be centered around. This reduces error in the approximation.

The distortion of the earth changes the direction of the vertical everywhere but at the equator and poles. Because of this a local vertical coordinate system, CS_LV, is under CS_I. This simplifies the kinematics when it's used in conjunction with the players coordinate system. Each CS_LV has its origin at the center of mass for each player. One axis points down the gravity vector, and another points in the local North direction. The other axis is 90° from the first two using the right hand rule convention.

Under the CS_LV is the body coordinate system (CS_B) which is attached to the physical airframe of each player and moves with the player. In other words it's attached to the players platform. Its origin is also at the center of mass, so relations between those two systems only involves rotations. The coordinate system with two players is shown on the following page in Figure 1. Each player has its own CS_LV and CS_B. To obtain positional information of the players with respect to one another, traverse the tree to the common coordinate system.

FIGURE 1



Another coordinate system which may be used is the atmosphere coordinate system (CS_A). This would be attached under the local vertical so wind could be conveniently described in local direction and velocity in relation to the earth.

Other coordinate systems can be used to simplify aerodynamic computations such as the stability axis system, CS_S, and the relative wind coordinate system. Both of these coordinate systems would be attached to the local vertical.

There are many coordinate systems below CS_B, and they are usually unique to the equipment used. Different gimbal systems for gyros require different combinations of coordinate systems for gyros, gimbals, platforms and rotation sensors. Another example is of a radar antenna scanning while it is moving in its gimbal system. In this case one coordinate system should be located at the antenna's pedestal and another should be aligned with the moving antenna. This way, antenna motion is encapsulated in the relationship between the pedestal coordinate system and the antenna system.

15. ***Issue:*** J-MASS uses a simplistic spherical model for the earth, whereas DIS uses an oblate ellipsoid model (WGS84 model). This causes problems when trying to reference a point on the surface of the earth model. For further analysis see Appendix A.

Reference: J-MASS Coordinate Systems Standards, (J-MASS-CSS-1.0) 16, August 1991.

Military Handbook on Datum, Projections, Grids and Common Coordinate Systems, Transformation of DoD (MIL-HDBK-600008). May, 1991.

Standard for Information Technology Protocols For DIS Applications. Version 1.2, 1993.

Disposition: J-MASS currently uses a perfect sphere to model the earth, and DIS uses the WGS84 model of the earth. The WGS 84 model is a much more accurate representation of the earth. In the near term, algorithms can be developed to map all the points on one model to another. This won't be difficult. The difficulty is that the distance between two points on the sphere must be the same as the distance between the two points after they have been mapped to the ellipsoid and vice versa. If the battles are contained within a small area on or near the equator there will be very little error, because the semi-major axis of the WGS84 model is approximately equal to the radius of the sphere J-MASS uses. (Semi-Major axis = 6,378,137 meters, Radius of sphere = 6,378,150 meters for an error of .0002%). The fastest and easiest short term method of obtaining very little error in the differences between the J-MASS and DIS model of the earth would be to have the battles contained in a small area centered around the equator. J-MASS needs to use the WGS84 model as soon as possible. Because the model is in an ADA package (ECP_Earth_Characteristics_Package) it shouldn't be too difficult to uses other models.

IST is currently working on a method of maintaining the distances between any two points on both models. This may involve using a large table of multiplication factors based on what area of the earth you are on. For example, if the battle takes place in Nicaragua you multiply the distances used by the factor found in the table. Because J-MASS does nearly all of the computing the distances should be converted on the DIS side.

In conclusion, Rue Hestand recommends changing J-MASS to the WGS84 model as soon as possible. This would eliminate all concerns. Until this is done the battles will be very accurate at or near the equator. If J-MASS can't have its model of the earth changed in a reasonable period of time it may be necessary to implement the conversion algorithms on which IST is currently (5/93) working. The added processing time that this will take must also be examined.

16. *Issue:* Vehicle position, relative to the earth, is computed differently in J-MASS and DIS. In DIS a WGS 84 ellipsoid is used to represent the earth. In the current implementation of J-MASS a spherical model is used to represent the earth (Issue 15). Both simulations use Cartesian coordinates to compute the location of the earth's surface and vehicles in the simulation. Problems arise in representing entities created in DIS in the J-MASS environment and vica versa. The problem relates to computing vehicle position relative to the earth and other vehicles. The computation of position is necessary to determine relative geometry between vehicles, crashes, navigation, weapons delivery, etc. A problem only arises because the earth is used as an intermediate datum for both systems.

Reference: Lin, Kuo-Chi and Ng, Huat Keng, "Interconversions Between Different Coordinate Systems", Institute for Simulation and Training.

Science Applications International Corp., Aeronautical Systems Operation-Dayton, OH Memorandum, dated 2 June 1993, Log WBM:062, from Barry Michael to Onda Simmons.

Disposition: Both the geodetic system and the geocentric system have a basis at the center of the earth. Therefore, if one were merely interested in computing relative geometry between two vehicles (without regard to the earth) then a simple conversion between spherical and Cartesian coordinates would suffice. However, as noted above, the location of the earth's surface relative to the coordinate system origin is important for many reasons. Therefore, the contour of the earth's surface is a matter which must be considered in the placement of vehicles in J-MASS and DIS. A mathematical description of the surface as represented by J-MASS and by DIS is necessary for simulation purposes. A mathematical technique to move consistently between J-MASS and DIS is necessary for Amherst's effort to create a DIS<->J-MASS interface.

For the sake of simplicity we will assume that the geodetic system and a spherical earth model are synonymous (in fact, though, the geodetic system is one of equal potential and, as such, may have slight perturbations due the heterogeneous composition of the earth). The geodetic system uses an offset from the radius for the earth, known as altitude, h, to compute position. The WGS84 earth is represented as an oblate ellipsoid. Suffice it to say that the relationship in distance between any given latitude, longitude, and elevation between these two systems is a complex mathematical relationship which is dependent upon latitude and longitude. The relationship is an implicit set of mathematical equations which have been previously documented by IST and mechanized in software. Therefore, in the referenced document the comment in paragraph 2 of the problem statement which refers to "No clean solution..." is not strictly true. A clean solution does exist. The solution, however, is complicated.

The situation described above is further complicated by the fact that linear vectors in Cartesian space are, in reality, great circle routes in the geodetic system and some other type of non-linear relationship in WGS84. In other words, flying at a constant altitude is thought of as being linear even though it's actually circular. Therefore, distances, velocities, and accelerations are all non-linear in form in WGS84 and are most easily referenced by spherical coordinates. These differences have not been noticeable for short duration missions or at speeds flown by most aircraft. No subjective or objective determination of when differences become noticeable have been made in any of the above areas for J-MASS or DIS.

IST believes that the transformations described by SAIC will work for demonstration purposes. However, the SAIC scheme has limited application in the long term because of the requirement for all simulators to make the same transformations. In addition, the simple process described by SAIC will most likely result in jerky vehicle movement in the DIS and J-MASS environments. The reason is that dead reckoning, without smoothing, will tend to cause vehicle displacements to be abrupt. The situation will be more evident as velocity (linear and angular) increases.

The conversion process between a spherical and WGS84 earth does not require transcendental functions. IST believes that Taylor or Maclauren series can be used to approximate transcendental functions. The exact solution for a conversion from geodetic to geocentric involves transcendental functions, but is relatively simple (Appendix A). Conversions from geocentric to geodetic are much more complicated involving iterative techniques (uses Newton-Raphson method for convergence) and transcendental functions (Reference 2). The computational savings are dependent on the degree of accuracy desired from the expansion.

One final comment is appropriate to the referenced document. The last page of the referenced document discusses limitations in fidelity. The following sentence should be inserted before the last sentence of the last paragraph of the section entitled Cons, "Also, interactions with terrain, high g maneuvers, and air to air engagements would suffer from lower fidelity calculations."

APPENDIX A

"An analysis of the error of J-MASS with respect to DIS concerning calculated distances between two points on the surface of their respective models"

Issue: As discussed in Issue 15, J-MASS uses a spherical model of the earth, and DIS uses an ellipsoid (WGS84). When calculating distance between two points knowing those points' latitude and longitude, J-MASS and DIS will obtain different results. How different are the results in the Hunter Ligget area?

Reference: Military Handbook on Datum, Projections, Grids and Common Coordinate Systems, Transformation of DoD (MIL-HDBK-600008). May, 1991.

Rapp, Richard H., "Geometric Geodesy, Part 1." Ohio State University - Department of Geodetic Science and Surveying. April, 1991.

J-MASS Coordinate Systems Standards, (J-MASS-CSS-1.0) 16, August 1991.

Disposition: The area of concern is latitude, 36°9' - 35°42' North, and longitude, 122°30' - 121°56' West. This area encompasses Fort Hunter Ligget (see enclosed map). Mr. Hestand found that the error in the distance between two points increased as a position vector constructed with those two points moves northward. When the vector points in a north/south direction, the error is very small (under 1%). When the vector points east/west, the error is large (23%-24%). The error measurement assumes the DIS distance is the accurate distance. The following pages show how the error was calculated. Refer to the enclosed map to see where the points being used are found.

First the given latitude and longitude was translated into the Cartesian coordinate system for the ellipsoid (WGS84).

$$X = (R_n + h) \cos\phi \cos\lambda$$

$$Y = (R_n + h) \cos\phi \sin\lambda$$

$$Z = \left(\frac{b^2}{a^2} R_n + h \right) \sin \phi$$

where ϕ is the latitude, λ is the longitude, h is the height above the surface, b is the semi-minor axis of the ellipsoid (b=6,356,752 m), a is the semi-major axis (a=6,378,137 m), and R_n is the radius of curvature in the prime vertical given by the equation below.

$$R_n = a^2 / \sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}$$

These equations can be found in MIL-HDBK-600008 (Reference 1).

All calculations assume the points are on the surface (h=0).

Point 1: $\phi = 35°42' = 35.7°$, $\lambda = 122°30' = 122.5°$

$R_n = 6.38542 \text{ X } 10^6$ m

$X = -2.78616 \text{ X } 10^6$ m

$Y = 4.3734 \text{ X } 10^6$ m

$Z = 3.70121 \text{ X } 10^6$ m

Point 2: $\phi = 35°42' = 35.7°$, $\lambda = 121°56' = 121.9333°$

$R_n = 6.38542 \text{ X } 10^6$ m

$X = -2.74277 \text{ X } 10^6$ m

$Y = 4.40074 \text{ X } 10^6$ m

$Z = 3.70121 \text{ X } 10^6$ m

Point 3: $\phi = 36°09' = 36.15°$, $\lambda = 121°56' = 121.9333°$

$R_n = 6.38558 \text{ X } 10^6$ m

$X = -2.72728 \text{ X } 10^6$ m

$Y = 4.375882 \text{ X } 10^6$ m

$Z = 3.74164 \text{ X } 10^6$ m

Point 4: $\phi = 36.15°$, $\lambda = 122.5°$

$R_n = 6.38558 \text{ X } 10^6$ m

$X = -2.7704 \text{ X } 10^6$ m

$Y = 4.3487 \times 10^6$ m

$Z = 3.74164 \times 10^6$ m

Point 5: $\phi = 35.7°$, $\lambda = 122°56' = 122.21667°$

$R_n = 6.38542 \times 10^6$ m

$X = -2.7645 \times 10^6$ m

$Y = 4.3871 \times 10^6$ m

$Z = 3.70121 \times 10^6$ m

Point 6: $\phi = 35.925°$, $\lambda = 122.21667°$

$R_n = 6.3855 \times 10^6$ m

$X = -2.7567 \times 10^6$ m

$Y = 4.37477 \times 10^6$ m

$Z = 3.72146 \times 10^6$ m

There are two ways to obtain the distance between any two points on the surface of an ellipsoid (a sphere is a special case of an ellipsoid with the two axis equal). One can use a linear approximation or find the arc length.

For the perfect sphere J-MASS uses, the arc length between two points on the sphere is given by Equation 1 below:

Equation 1 $$D_{p1p2,J} = r\Delta\theta$$

where r is the radius of the sphere (r=6,378,150m), and $\Delta\theta$ is the angle between the two points. A linear approximation can be made by using the Law of Cosines, (with two sides being equal to r).

$$D_{p1p2,J} = 2r^2 - 2r^2 \cos(\Delta\theta)$$

Because using the arc length is more accurate, it was used to calculate distance. The linear approximation for a sphere was given for completeness.

For the ellipsoid model DIS uses, calculating the arc length is more involved. Calculating the distance between two points on a constant latitude line (λ changes) is shown below in Equation 2.

Equation 2 $D_{p1p2,D} = N\cos\phi(\lambda_2 - \lambda_1)$

where $N = [a / (1 - e^2\sin^2\phi)^{1/2}]$ and $e = [(a^2 - b^2)/a^2]^{1/2}$ = eccentricity

The distance between two points on a constant longitude line is shown below in Equation 3.

Equation 3 $D_{p1p2,D} = a(1 - e^2)[A(\phi_2 - \phi_2) - B/2(\sin2\phi_2 - \sin2\phi_1) +$

$C/4(\sin4\phi_2 - \sin4\phi_1) - D/6(\sin6\phi_2 - \sin6\phi_1) + ...]$

The remaining terms are neglected since they're close to zero. For the WGS84 model the constants are shown below.

$A = 1.0050526$ $B = 5.06318 \times 10^{-3}$

$C = 1.06279 \times 10^{-5}$ $D = 2.08181 \times 10^{-8}$

Equations 2 and 3 and the equations for the constants are derived in Reference 2 ("Geometric Geodesy, Part 1"). A linear approximation is shown below in Equation 4,

Equation 4 $D_{p1p2} = [(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2]^{1/2}$

Now I will calculate the distances between points on both the J-MASS sphere and the DIS ellipsoid.

Distance between Points 1 and 2.

$\phi = 35.7^\circ$, $\lambda_2 = 121.933^\circ$, $\lambda_1 = 122.5^\circ$, $h = 0$

J-MASS $\Rightarrow$ $D_{12,J} = 6.30815 \times 10^4$ m (using Equation 1)

DIS $\Rightarrow$ $D_{12,D} = 5.12856 \times 10^4$ m (using Equation 2)

Therefore, there is a difference of 11.8 km or

Percent Error = $|D_{12,J} - D_{12,D}| / D_{12,D} * 100\% = 23.0\%$ error

Using Equation 4 for the linear approximations, we obtain for DIS:

$D_{12,D} = 5.128516 \times 10^4$ m

The error when using the approximation is 8.57×10^{-4} % error. Thus if necessary using a linear approximation for distances less than 100 km will result in very little error.

Distance between Points 1 and 4.

$\phi_2 = 36.15^\circ$, $\phi_1 = 35.7^\circ$, $\lambda = 122.5^\circ$, $h = 0$

J-MASS $\Rightarrow$ $D_{14,J} = 5.00938 \times 10^4$ m (using Equation 1)

DIS $\Rightarrow$ $D_{14,D} = 4.99309 \times 10^4$ m (using Equation 2)

Therefore, there is a difference of 162.9m or .326% error.

Distance between Points 1 and 5.

$\phi = 35.7^o$, $\lambda_2 = 122.21667^o$, $\lambda_1 = 122.5^o$, $h = 0$

J-MASS $\Rightarrow$ $D_{15,J} = 3.15406 \times 10^4$ m (using Equation 1)

DIS $\Rightarrow$ $D_{15,D} = 2.5643 \times 10^4$ m (using Equation 2)

Therefore, there is a difference of 5,897.6m or 23.0% error.

Distance between Points 3 and 4.

$\phi = 36.15^o$, $\lambda_2 = 121.9333^o$, $\lambda_1 = 122.5^o$, $h = 0$

J-MASS $\Rightarrow$ $D_{34,J} = 6.30815 \times 10^4$ m (using Equation 1)

DIS $\Rightarrow$ $D_{34,D} = 5.0995 \times 10^4$ m (using Equation 2)

Therefore, there is a difference of 12,087 m or 23.7% error.

Distance between Points 1 and 6. Here the linear approximation is used because the points don't lie on a constant latitude or longitude line, and I've shown the difference is small.

$\phi_2 = 35.925^o$, $\phi_1 = 35.7^o$, $\lambda_2 = 122.21667^o$, $\lambda_1 = 122.5^o$, $h = 0$

J-MASS $\Rightarrow$ $D_{16,J} = 4.0276 \times 10^4$ m (using Equation 1)

Using the Cartesian points for 1 and 6, the distance is:

DIS $\Rightarrow$ $D_{16,D} = 3.57747 \times 10^4$ m (using Equation 4)

Therefore, there is a difference of 4,501.4m or 12.58% error.

By comparing the error from 3 to 4 with the error from 1 to 2, one can see the error increases as you approach the poles. At the equator the error is very close to zero no matter which direction the position vector is facing. At Fort Hunter Ligget, the error is very large if the position vector made from the two points is in an east/west direction. If the vector runs north/south the error is very small. This large difference in error being dependent on the orientation of the position vector is going to make it very difficult to correct. Every time a distance must be computed J-MASS would need to know the latitude and orientation of the position vector. J-MASS could then look at a huge factor table based

on that information, and then multiply it's distance by the factor found in the table. This seems to be impractical, because the table would have to be very large and all though that method would reduce the error considerably there would still be appreciable error in some cases. There may be an algorithm to convert the spherical model to the WGS84 model, but I haven't seen or heard about an algorithm which performs that operation while maintaining distance relationships.

This analysis was performed at sea level (h=0). As the height above sea level increases so will the error, which must be taken into consideration. The correction table would then have a third variable based on height above sea level.