Beware the real export compliance challenges of simulation software

The benefits of simulation software can be very clear but the inherent export control implications not so. Jeffrey Richardson examines the real challenges of virtualisation.

Simulation software plays a vital role within industry by virtualising the natural trial and error of research and development, as well as reducing the expensive hands-on training time necessary to achieve mastery over complex machines. Simulation software may be used for everything from manipulating a simulated crew station that prepares pilots to operate a JSF F-35, to creating the military scenarios necessary to support combat training sessions. Moreover, simulation software may be used for industrial modelling necessary to test and develop certain end-items or components, such as a gas turbine engine within an aircraft or a transmission within a ground vehicle. Simulation software may be subject to U.S. export controls when designed for either military training or industrial modelling, depending upon the simulation software's performance characteristics, utilised technical data, and simulated end-item.

As to military training simulation software, on 1 July 2014, U.S. export control reform reached military training equipment, transferring controlled items from the United States Munitions List ('USML') of the International Traffic in Arms Regulations ('ITAR') to the Commerce Control List ('CCL') of the Export Administration Regulations ('EAR'). Specifically, the classification of certain military training items were transferred from USML Category IX to Export Control Classification Number ('ECCN') 0A614 (both classifying military training equipment).[1] Now both USML Category IX and ECCN 0A614 include traditional military training equipment, and provide for the classification of simulators and simulation software.

As a result of U.S. export control reform, dependent classifications under both the USML and CCL are required to determine the export control classification of certain military training items. A dependent classification occurs when the export control classification of a simulator (and simulation software) is dependent upon the export control classification of a simulated end-item.

Dependent classifications were common to the USML prior to export control reform. For example, if an aircraft was classified under the pre-reform USML Category VIII, then the parts and components designed for that aircraft were also classified under USML Category VIII.

In the era of post-reform, classification of simulators is often dependent upon the classification of the simulated end-item. By way of example, to determine if a crew station simulator is classified under USML Category IX, one must first determine if the simulated end-item (i.e., the particular crew station) is classified under the USML. Similarly, to determine if the simulator is classified under ECCN 0A614, one must first determine if the simulator is classified

under USML Category IX, which, in turn, requires determining if the simulated end-item is classified under the USML.

In many cases, industrial modelling simulation software is EAR99 because it simply provides a mathematical output upon processing the provided data sets and is not identified on the CCL.[2] By contrast, however, certain industrial modelling simulation software is independently classified by reason of the performance of the simulated end-item, while other industrial modelling simulation software is classified based upon a particular functionality designed into the software. Moreover, the use of controlled technical data within industrial modelling simulation software can create an export control violation risk for both the end-user and the software developer when technical support services are provided. Often controlled technical data from the end-user must be configured by the software developer to permit processing by the industrial modelling simulation software, increasing the potential risk for a deemed export or the unexpected provision of defence services violating the ITAR.

This article provides (i) an overview of the U.S. export control classifications for military training simulation software under USML

Category IX and ECCN 0A614, as well as industrial modelling simulation software classified elsewhere on the CCL; (ii) an analysis of the problems arising from dependent classifications for simulation software, including concerns arising from providing technical data during the technical support process; and (iii) strategies for export control compliance personnel to cope with simulation software.

U.S. export control classification
Military training simulation software may be classified under the ITAR's USML and EAR's CCL, while industrial modelling simulation software is most often EAR99 or classified on the CCL.

USML classification of military training simulation software
To understand the classification of military training simulation software, one must begin with a review of the Military Training Equipment and Training listed on USML Category IX.

- USML Category IX(a) classifies training equipment for ground, surface, submersible, space, or towed airborne targets that (i) mimic a specific defence article, or (ii) provide hit/miss performance information for defence articles.
- USML Category IX(b) classifies simulators (i) that replicate the operation of an individual crew station, mission system, or a weapon of a USML end-item, as well as (ii) *software* and associated databases that can be used to model or simulate the following: trainers listed on USML Category IX(a), battle management, military test scenarios/models, or the effect of weapons listed on the USML. [3]

To elaborate on the USML simulators, first, a crew station is the cockpit or area where crew members are situated during the operation of the defence article. Second, 'mission systems' are systems that are defence articles that perform specific military functions, such as providing military communication, electronic warfare, target designation, surveillance, target detection, or sensor capabilities. Third, the simulation of a weapon of a USML end-item, by example, would include simulating the trajectory of a torpedo (weapon) launched from a submarine (USML end-item).

USML Category IX(e) classifies

technical data and defence services that directly relate to defence articles listed on USML Category IX(a) classifying trainers and IX(b) classifying simulators.

CCL classification of military training simulation software
After confirming the military training simulation software is not military training equipment and training listed on USML Category IX, one may continue the classification of the military training simulation software by reviewing ECCN 0A614 (for military training equipment) and ECCN 0D614 (for corresponding software). Specifically, ECCN 0D614 classifies software specially designed for the development, production, operation, or maintenance of commodities controlled by ECCNs 0A614 or 0B614. ECCN 0A614.a classifies equipment specially designed for military training that is not listed on USML Category IX, while ECCN 0A614.x classifies parts, components, accessories, or attachments that are specially designed for a commodity controlled under ECCN 0A614 or an article listed on USML Category IX. And, ECCN 0B614 classifies test, inspection, and

production equipment for ECCN 0A614 (military training equipment).

For most 600 series ECCN designations a .y classification is available for lightly controlled items subject only to anti-terrorism ('AT') controls. Conversely, however, there is no .y classification for ECCN 0A614 or 0B614. Consequently, there is no .y classification for ECCN 0D614 which classifies software related to military training equipment. Because there is no .y classification available, the AT reason for control is simply not available for ECCN 0D614 (software for military training equipment).

Classification of industrial modelling simulation software
By nature, industrial modelling simulation software usually offers a blank canvas dependent upon

imported data-sets to provide the end-user with the desired industrial modelling analysis. The end-user often supplies relevant data sets or configures the software to achieve the desired industrial modelling analysis. Apart and separate from its corresponding data sets, industrial modelling simulation software is often EAR99, but there are certain circumstances when the industrial modelling simulation software itself is listed on the CCL.

In contrast to EAR99 industrial modelling simulation software, certain industrial modelling simulation software is independently controlled based upon a particular functionality designed into the software. By way of example, ECCN 9D004.g.1 classifies simulation software specially designed to predict aero thermal, aeromechanical and combustion conditions in aero gas turbine engines, while ECCN 9D004.g.2 classifies theoretical modelling predictions of the aero thermal, aeromechanical and combustion conditions, which have been validated with actual turbine engine (experimental or production) performance data. A software developer creating this type of

> *In contrast to EAR99 industrial modelling simulation software, certain industrial modelling simulation software is independently controlled based upon a particular functionality designed into the software.*

aerospace industrial modelling simulation software is well advised to understand the effect of these unique narrow classifications, and their corresponding controls on the ability to freely export the software to foreign customers.

Additionally, industrial modelling simulation software may be controlled due to the attributes of the simulated end-item. By way of example, ECCN 9D001 (software for certain aero gas turbines engines) controls software specially designed or modified for the *development* of equipment or technology controlled by ECCN 9A001 (certain aero gas turbines engines). As defined by the EAR, development relates to all stages prior to serial production, such as design, design research, design analyses, design concepts, assembly and testing of

prototypes. Specifically, ECCN 9A001 controls aero gas turbine engines that utilise specific technology, including for example specific gas turbine blades, vanes or tip shrouds, made from directionally solidified ('DS') or single crystal ('SC') alloys and having a stress-rupture life exceeding 400 hours at 1,273 K (1,000 °C) at a stress of 200 MPa. So, if the industrial simulation software is used for the development of an aero gas turbine engine within ECCN 9A001, then the industrial simulation software itself is classified and controlled under ECCN 9D001. This is an example of where the software itself is controlled due to the attributes of the simulated end-item for which the software was developed.

The problem of dependent classifications

The problem of dependent export control classifications is particularly acute within the world of simulation software.

The first type of dependent classification occurs when the simulation software classification depends upon the classification of a separate and distinct end-item. This may result if a software developer receives controlled technical data for an end-item with a different classification than the developer's

simulator, i.e. technical data from a USML Category VII ground vehicle may be received to run the simulation of a USML Category IX(b)(1) crew station simulator.

A second type of dependent classification occurs when the software developer must consider the classification for end-item technical data that is provided to run the simulation, because the technical data received to run the industrial modelling simulation may have different controls from the industrial modelling simulation software. For example, a software developer may receive technical data from a USML Category XIX gas turbine engine such as the GE38, AGT1500, CTS800,

TF40B, T55, T60, and T700, or technical data listed on ECCN 9E003.a to make an aero gas turbine engine with parts comprised of organic composite materials designed to operate above 588 K.

Dependent classifications affecting USML items

Cases of dependent classifications for USML Category IX(b)(1) may arise when a software developer produces software for a simulator that replicates the operation of a controlled (i) crew station, (ii) mission system, as well as (iii) the operation of a weapon for a USML controlled end-item. To classify such a simulator, the software developer must know in detail the export control classification of the controlled crew station, mission system, or the weapon for a USML controlled end-item. For example, if the simulator replicates a crew station for a ground vehicle, the software developer must know the classification of that ground vehicle.

The software developer may lack sufficient information to properly classify the controlled ground vehicle to be replicated by the simulator. Depending on the simulated weaponry or armour, the replicated crew station could be based on a ground vehicle listed on USML Category VII or ECCN

A simulated end-item classified under a USML category of the ITAR correspondingly causes the simulator to be classified under USML Category IX.

0A606 (both classifying ground vehicles). Alternatively, a ground vehicle such as a general purpose HMMWV may be classified EAR99 unless up-armoured. In sum, these cases require the classification of the simulated end-item drives the classification of the simulator, while the detail necessary to classify the simulated end-item may simply not be provided to the software developer.

Dependent classifications affecting CCL Items

Similarly, a dependent classification is required to complete the export control classification under ECCN 0A614 (for military training equipment) and ECCN 0D614 (for corresponding

software). Here, the software developer must first confirm whether the end-item is specially designed for military training, but not otherwise listed on USML Category IX, because ECCN 0A614 classifies equipment specially designed for military training that is not listed on USML Category IX. The critical information necessary to confirm whether the simulator is listed on USML Category IX is the classification of the simulated end-item. In short, a simulated end-item classified under a USML category of the ITAR correspondingly causes the simulator to be classified under USML Category IX. Simulated end-items not otherwise listed on the USML force a dependent classification under ECCN 0A614.

For example, to classify the software for an aircraft crew station simulator, the software developer must confirm whether the simulated end-item aircraft is listed on USML Category VIII, which may include: (1) Bombers; (2) Fighters, fighter bombers, and fixed-wing attack aircraft; (5) Unarmed military unmanned aerial vehicles ('UAVs'), as well as any (i) U.S.-origin aircraft that bear an original military designation of A, B, E, F, K, M, P, R, or S; or (ii) Foreign-origin aircraft specially designed to provide functions equivalent to those listed aircraft. This classification review requires the software developer possess sufficient detail to confirm whether the simulated end-item aircraft is listed on USML Category VIII. If the simulated end-item aircraft is not listed on USML Category VIII, then (i) the aircraft crew station simulator is not listed on USML Category IX, and (ii) the software developer may proceed to the classification analysis under ECCN 0A614 (for military training equipment) and ECCN 0D614 (for corresponding software). The classification of the simulated end-item drives the classification of the simulator and simulation software, while the detail necessary to classify the simulated end-item may not be provided to the software developer.

ITAR defence services and EAR99 simulation software

A software developer may in fact develop simulation software that is classified EAR99. But running complex simulator software often requires some technical assistance for the end-user by the developer. Often an end-user

reaches out to a developer to configure the end-user's data sets used to run the desired simulation, or interpret and apply the results from a simulation. Prior to doing this, the software developer must confirm that the end-user's technical data enabling the simulation is not ITAR-controlled technical data.

Under the ITAR, the transfer of controlled technical data may result in the software developer unintentionally providing defence services to end-users because such assistance may constitute either (i) furnishing assistance to foreign persons with the design, development, engineering, manufacture, production, assembly, testing, repair, maintenance, modification, operation, processing or use of defence articles, or (ii) furnishing to foreign persons of any technical data controlled under the ITAR. This may be difficult for the software developer to discern, because the simulation software processes the data in the same manner with the resulting data output looking the same, irrespective of whether the end-user's technical data set input is ITAR-controlled or merely EAR99.

This underscores the problem that the software developer is wholly reliant upon the software end-user's assessment of the export control classification to determine whether the technical data shared back and forth is controlled under the ITAR. Given this reliance, the software developer must consider the potential ramification of unintentionally providing a defence service when providing technical support. The software developer has no inherent need to differentiate the data sets. Indeed, the software developer simply processes the configured data sets without any colouring as to the U.S. export control classification, and no independent need to distinguish whether an ITAR-controlled end-item generates the data sets. The software developer must be vigilant while providing technical support for data-dependent industrial modelling simulation software, so as not to provide a defence service when advising as to the configuration of ITAR-controlled technical data.

600 series technical data and EAR99 simulation software

Similarly, the software developer may receive 600 series end-user data while providing technical support. For example, the end-user's data set for a simulated end-item may be classified ECCN 0E606 (ground vehicle technology) as technology required for the development, production, operation, installation, maintenance, repair, overhaul, or refurbishing of ground vehicles and related commodities in 0A606. Again, in this case, the software developer is wholly reliant upon the software end-user's assessment of the export control classification to determine whether the end-user's data set is controlled under the 600 series. The software developer may decide to trust the classification information provided by the end-user, decline the simulation software sale, or attempt to independently verify the export control classification. Often, however, the software developer lacks both relevant information and access to the personnel necessary to undertake an effective independent inquiry as to the U.S. export control classification of

the simulated end-item. The software developer must be vigilant while providing technical support for data-dependent industrial modelling simulation software, so as not to permit a deemed export when processing the ITAR- or EAR-controlled technical data provided by the industrial modelling simulation software end-user.

Compliance strategies for simulation software

To better implement export compliance, the software developer may seek to (i) acquire additional information about the nature of the data inputs and end uses from the end-user to assist with the export control classification, or (ii) subject end-users to an end-user licence agreement ('EULA') that limits software use to permitted end uses and use with non-controlled data.

Acquire classification information from the end-user

The software developer may seek additional classification information from the end-user to perform an export control classification for the simulated end-item. This information may include manufacturing specifications, performance capabilities, or supporting technical data. If accurate, the additional classification information may assist the software developer in implementing appropriate compliance pertaining to the end-user's project. But the software developer must carefully vet the information provided by the end-user, as problems arise when the end-user lacks sufficient classification information or analysis expertise. These problems may result in (i) both overly-stringent classifications and kneejerk 'not controlled' classifications, as well as (ii) the inability to accurately complete dependent classifications.

At the onset of U.S. export control reform, DDTC registrants subject to the ITAR initially seemed satisfied to continue to state that all technical data

Under the ITAR, the transfer of controlled technical data may result in the software developer unintentionally providing defence services to end-users.

provided for use with simulation were classified under the ITAR. These broad statements applying blanket ITAR classifications were made so the DDTC registrant could avoid the complex and granular analysis required to classify their technical data under the new 600 series of the EAR. In many cases, because the DDTC registrant was a U.S. company with few or no foreign national employees, and the end-item was delivered to a U.S. company wholly in the U.S., the nuanced 600 series classification was not beneficial – despite the less restrictive export controls. On the other hand, many end-users myopically state that all data inputs, simulated end-items, and end uses are not controlled when such conclusions are unfounded and incorrect. So the software developer should be careful to watch for the end-users who may incorrectly apply either overly-stringent classifications or kneejerk 'not controlled' ITAR classifications to their technical data.

In particular for dependent classifications, the software developer should be wary because the

classifications by the end-user may be wrong, no matter how thoughtfully done. The software developer must seek additional background information to confirm the accuracy of dependent classifications completed by the end-user. This confirmation may include requests for additional information describing the end-item, discussing the end use in detail with the end-user's customer, and independently confirming key technical specifications that are the determinative elements for the export control classification.

Restrict use of the simulation software

A software developer may choose to restrict the use of the simulation

software through the EULA. Common EULA restrictions require the end-user to comply with all U.S. export control laws, not re-export the software to a prohibited destination or party, and not use the software for any prohibited end use. These restrictions may be directly stated within the EULA licence requirement grants, so the end-user agrees to comply with U.S. export controls upon accessing the simulation software.

If the software developer has a particular concern about an end-user's use of the simulation software that could violate U.S. export controls, then the software developer may request the end-user sign a destination control statement with enhanced end use and end-user restrictions. A signed statement demonstrates a more explicit and proactive commitment by the end-user not to use the simulation software in violation of export controls. The signed statement would prohibit (i) re-export of simulation software to a prohibited destination or anyone appearing on the U.S. consolidated screening list, and (ii) restricted end

uses such as nuclear or for chemical and biological weapons.

Conclusion

Through virtualisation, simulation software provides economic efficiencies for training as well as research and development. Export compliance personnel should exercise caution with simulation software, as concerns naturally arise regarding interrelated classifications and the exchange of technical data during the end-user support process. With diligence, export compliance personnel can learn to cope with the many sides of simulation software.

Jeffrey Richardson is a senior attorney in the Troy office of Miller Canfield.

richardson@millercanfield.com