Simulation-Supported Wargaming for Analysis of Plans

Conference Paper · October 2015

8 authors, including:

Jo Erskine Hannay
Forsvarets Forskningsinstitutt

Vegard Kvernelv
Forsvarets forskningsinstitutt

Jens Inge Hyndøy
Forsvarets Forskningsinstitutt

Simulation-supported Wargaming for Analysis of Plans

Solveig Bruvoll, Jo E. Hannay, Guro K. Svendsen, Martin L. Asprusten, Kjell Magne Fauske, Vegard B. Kvernelv, Rikke A. Løvlid, Jens Inge Hyndøy
FFI – Norwegian Defence Research Establishment
P.O. Box 25, NO-2027 Kjeller
NORWAY

solveig.bruvoll@ffi.no jo.hannay@ffi.no guro.svendsen@ffi.no martin.asprusten@ffi.no
kjell-magne.fauske@ffi.no vegard.kvernelv@ffi.no rikke.lovlid@ffi.no
jens-inge.hyndoy@ffi.no

ABSTRACT

Wargaming is used in the military decision making process to visualize the execution of a preliminary plan or course of action in order to analyze and discover weaknesses and possibilities. The wargaming is traditionally done manually on a paper map, and the course of events is determined based on the experience and assumptions of the officers conducting the wargame. This paper describes ongoing research in Norway on the development of a demonstrator for Simulation-supported Wargaming for Analysis of Plans – SWAP. The focus is particularly on the synchronization of cooperating and supporting units, aiming to enable the planning group to more easily distribute supporting units to its subordinates when the support is most needed. This tool is intended to integrate simulation-supported wargaming in the planning process and thereby increase the quality of plans and decrease the planning time. SWAP uses a computer generated force federated with an agent-based simulation of C2 and combat management for simulation of a plan. It takes as input elements of a brigade plan from the Norwegian Command and Control Information System (C2IS) (NORCCIS). A web-based tool has been developed to support the officers in creating a synchronization matrix and to review the results of the simulation. The user can follow the simulated execution of the plan in the C2IS and receive information, such as fuel and ammunition consumption and casualties on both sides. C2 to Simulation (C2SIM) standards and a service-based approach is used to promote interoperability, while the simulation comprises a time-managed High Level Architecture (HLA) federation.

1.0 INTRODUCTION

During the planning of a military operation, the Operation Planning Group (OPG) first determines a Course of Action (COA) before refining the plan. At each planning step, the group considers several possible enemy COAs and alternative ways to execute the operation for each case. This analysis, combined with manual wargaming on a wargaming table, is done based on the experience of the officers in the OPG. In this paper we describe how simulations can be used to support this part of the planning process. Simulation-based wargaming is not intended to replace the analysis of experienced officers, but to provide more information to the officers' analysis.

The use of digital plans for military operations is increasing. Integration of simulation tools with the systems for digital plans, provides a convenient solution for the officers to perform simulation-based wargaming of their plans. Our work has focused on integrating a simulation system with the Norwegian Command and Control Information System (C2IS) NORCCIS. As a part of this, we have considered what type of information simulations are best suited to provide to the military officers.

We have developed Simulation-supported Wargaming for Analysis of Plans (SWAP), a demonstrator that simulates the execution of a digital plan. The simulation system has a distributed architecture that follows service-oriented principles, which enables the system to be easily configured, also to other application areas. Existing standards have been used, to make it easier to replace components of the system and encourage reusability. This paper describes the functionality of the SWAP system (Sec. 4), including its development (Sec. 5) and system architecture (Sec. 6).

2.0 RELATED WORK

Simulation support for planning may be divided into *statistical* and *case-driven* approaches [1]. A clear distinction should be kept as they have different objectives and areas of validity. Statistical approaches aim at finding statistically significant answers regarding the most likely outcome [2], [3]. Case-driven approaches on the other hand provide one possible outcome, for a selected case. Transparency and representativeness is therefore important in case-driven simulations, in which training of the cognitive understanding is a key aspect. The FRA simulation system Aide à la PLanification d'EngagementTactiques (APLET) is an example of a case-driven simulation, developed to support the planning process of the FRA brigade [4]. APLET takes as input a plan with battalion tasks either from the FRA C2IS SICF or a dedicated user interface. The execution of the tasks is simulated at the battalion level.

As the purpose of SWAP is to train and support, not to replace, the decision maker, it is a typical example of a case-driven simulation tool. As in conventional wargaming, it is up to the decision maker to evaluate and analyze the strengths and weaknesses of the plan including the robustness. Support to facilitate such analysis by providing additional information is a key element of SWAP. In [1], the authors suggest that case-based approaches can be further characterized by (1) *Increased exposure to cases*, in which simulations simply enable planners to investigate a larger number of COAs, (2) *Critical cases*, in which COA simulations are devised in order that planners may generalize results from one COA simulation to others according to certain heuristics, and (3) *Deliberate practice targeted at adaptive thinking*, in which simulations are devised aimed at boosting the cognitive skills of the planner. SWAP supports the first two of these approaches, but may in the future also be able to support the third approach.

In terms of Decision Support Systems (DSS), SWAP can further be classified as a *model-driven* DSS [5]. In terms of how the user interacts with the DSS, it is a passive DSS [6], in the sense that the system does not supply solutions, but offers support to the decision maker in his or her process of making a decision.

FFI has worked on developing autonomous simulations of military operations integrated with a C2IS for several years. Since 2005, FFI has participated in NATO science and technology research groups concerning development of a standard for a Coalition Battle Management Language (C-BML). In 2012 FFI developed a demonstrator for land operations consisting of a multi-agent system (MAS) and Computer-Generated Forces (CGF) [7], [8]. The MAS received C-BML orders from a C2IS and interpreted them into low-level BML tasks, which were executed by the CGF. Based on this work, a demonstrator for maritime operations was developed in 2014. The simulated entities had a high level of autonomy and were able to automatically perform advanced tasks such as searching for, and handling, enemy vessels according to defined rules of engagement. This functionality was intended to support Lower Control (LOCON) during a Computer Aided eXercise (CAX), enabling each operator to operate a larger number of entities. The development of SWAP is a continuation of this work [9].

3.0 THE PLANNING AND DECISION PROCESS

The NOR Army Planning and Decision Process (PBP) [10] prescribes a planning process for national brigade, battalion and company/squadron levels (Fig. 1). The process is aligned with the NATO Comprehensive Operations Planning Directive (COPD). Although the latter focuses on the NATO strategic and operational levels, its structure is intended to be used at lower levels as well.

In line with the COPD, the NOR PBP is divided into stages. During stages 1 and 2, a high-level outline of the operation, the Concept of Operations (CONOPS), is developed involving all relevant levels of command in an integrative and parallel manner. In stage 3, the CONOPS is used as a basis for developing a number of Courses of Action (COA); i.e., foreseen sequences of actions to be performed by BLUEFOR in various situations, are designed and evaluated, with input from supporting units (intelligence, engineering, logistics, etc.). Traditionally, COAs have been played out and visualized by wargaming manually on a wargaming table. Through evaluating and analyzing possible COAs, the CONOPS is refined (Stage 4) into an Operations Plan (OPLAN), which is eventually executed.

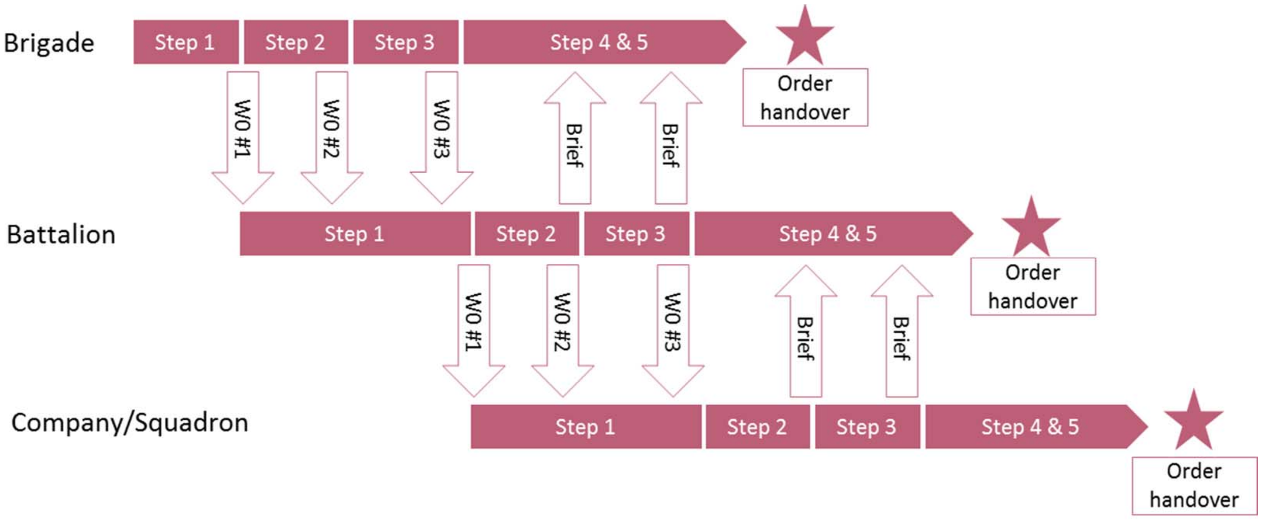

Figure 1: NOR Army Planning and Decision Process [10]

Main actors in the PBP are the Joint, Ground and Subordinate planning and intelligence officers (J3, J5, J2, G3, G5, G2, S3, S5, S2) who collaborate with other staff at their respective levels in an Operations Planning Group (OPG).

SWAP targets the COAs Development, Analysis and Comparison sub-processes under Step 3 of the PBP for brigade and battalion levels. As mentioned above, our focus is on simulation as a case-based decision support tool. In this sense, our proposition is that simulation support for wargaming enables one to:

- P1: develop, wargame, and analyze *more* COAs than with manual methods
- P2: get a clearer understanding of the possible consequences of a given proposed COA
- P3: understand more thoroughly the factors that underlie events triggered in a COA

4.0 SWAP FUNCTIONALITY

The objective of SWAP is to provide simulation support to the NOR PBP in accordance with propositions P1-P3. This support is provided to the OPG by means of a web-based user interface (UI) to SWAP. The SWAP UI lets the user import, refine, simulate and analyse COAs. The UI is also intended to provide examples on how a C2IS could be extended to provide simulation-based information supporting the OPG in the planning process.

The user interaction with SWAP has three mains aspects: controlling, predicting and describing. The user interaction workflow is visualized in Fig. 2.

The controlling aspect of SWAP involves the generation/modification of a plan, and submission of this plan to the simulation system for execution. At present, initial plans are developed using the planning module of the C2IS and exported to SWAP in Military Scenario Definition Language (MSDL) [11] format for further refinement, simulation and analysis. Tasks may also be added separately to refine the plan as the simulation evolves. The user interface for creating plans is inspired by the current C2IS, and task-definitions are formulated using C-BML [12] concepts.

The predictive support of SWAP can be used without starting up the simulation system. SWAP can be used in this mode both in an initial step before submitting the plan to the simulation system, or during simulation as the plan is executed. Support to improve synchronization of tasks is emphasized. We propose to integrate a tool directly into the C2IS, with an automatic coupling between the plan and a synchronization matrix. This is demonstrated in the UI of SWAP.

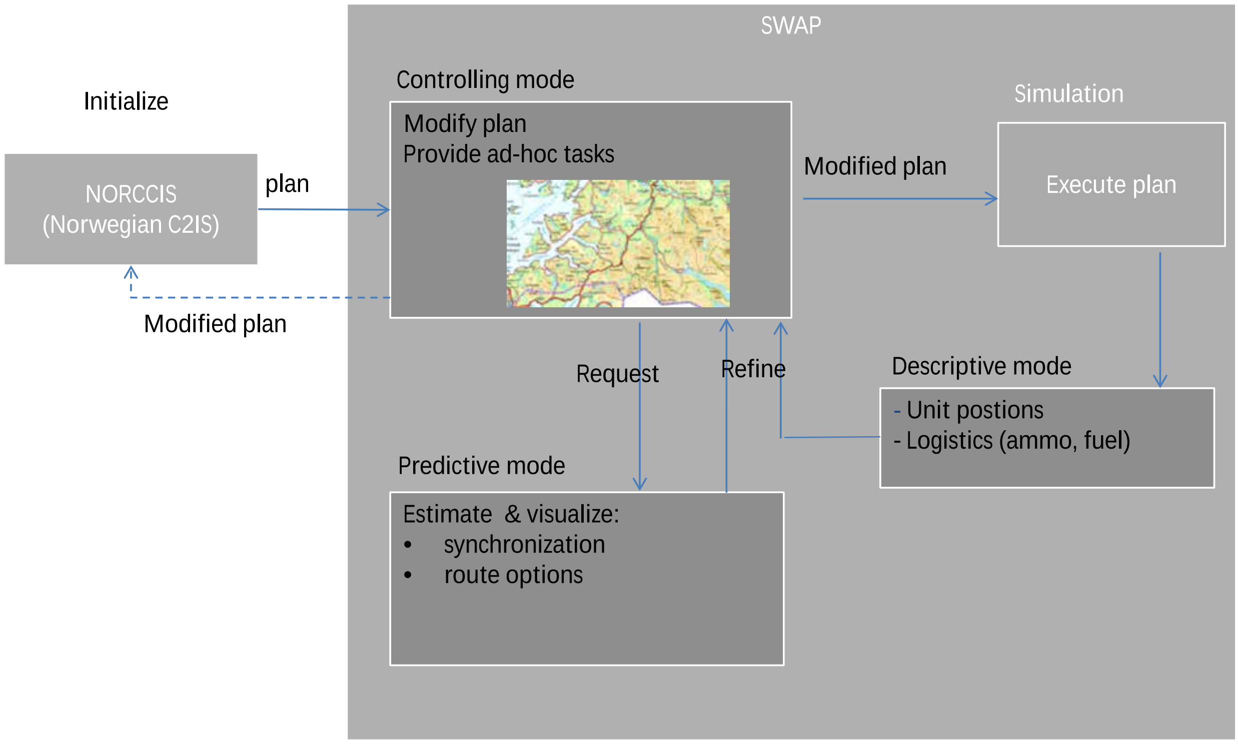

Figure 2: SWAP workflow from the OPG point of view. In an initial step, the plan is initiated using a C2IS. The plan is then modified (controlling mode), analyzed (predicitve mode) and executed, with simulation output presented in a description mode. Refinement of the plan, can be performed at any stage, as can the requests for predictions regarding timing and route options. On a longer term, SWAP should be fully integrated into the C2IS, such that modified plans can be reloaded (dotted arrow).

The synchronization matrix is a powerful tool to visualize how tasks given to the units should be synchronized with respect to each other, and an efficient means to communicate the plan. An automated synchronization matrix will save time during planning and also ensure that there are no discrepancies between the plan and the synchronization matrix. Further, the OPG should not spend more time than necessary estimating how much time a unit should use on movement from A to B. A route planning service (Sec. 6.5) is used by the simulation system to simulate movement, the same service can also be used in a predictive mode from the UI, to estimate how much time a certain unit would use on such movement. The OPG can set the timing for a movement task either manually or by applying the route planning service to obtain a time estimate. The timings can later be adjusted directly in the synchronization matrix.

In order for the OPG to know whether or not to trust the route planner, it is important with transparency, to clearly show the choices made therein, allowing the OPG to override if necessary. A separate view in the UI is therefore designated to provide a detailed visualization of routes planned by the route planning service, with respect to e.g. estimated speed and threat along the route. A screen shot from the UI demonstrating a route visualization is given in Fig. 3.

When a plan is submitted to the simulation system, and the simulation is started, the descriptive aspect is the most central part of the UI. It visualizes the execution of the plan and provides feedback on the current status. Based on the descriptions, the OPG may choose to refine the plan as the simulation evolves. The descriptive mode provides a view of the situation as unit symbols on a map as the simulation evolves. For an OPG, it is also important to see the current status of the units. This information is provided in a

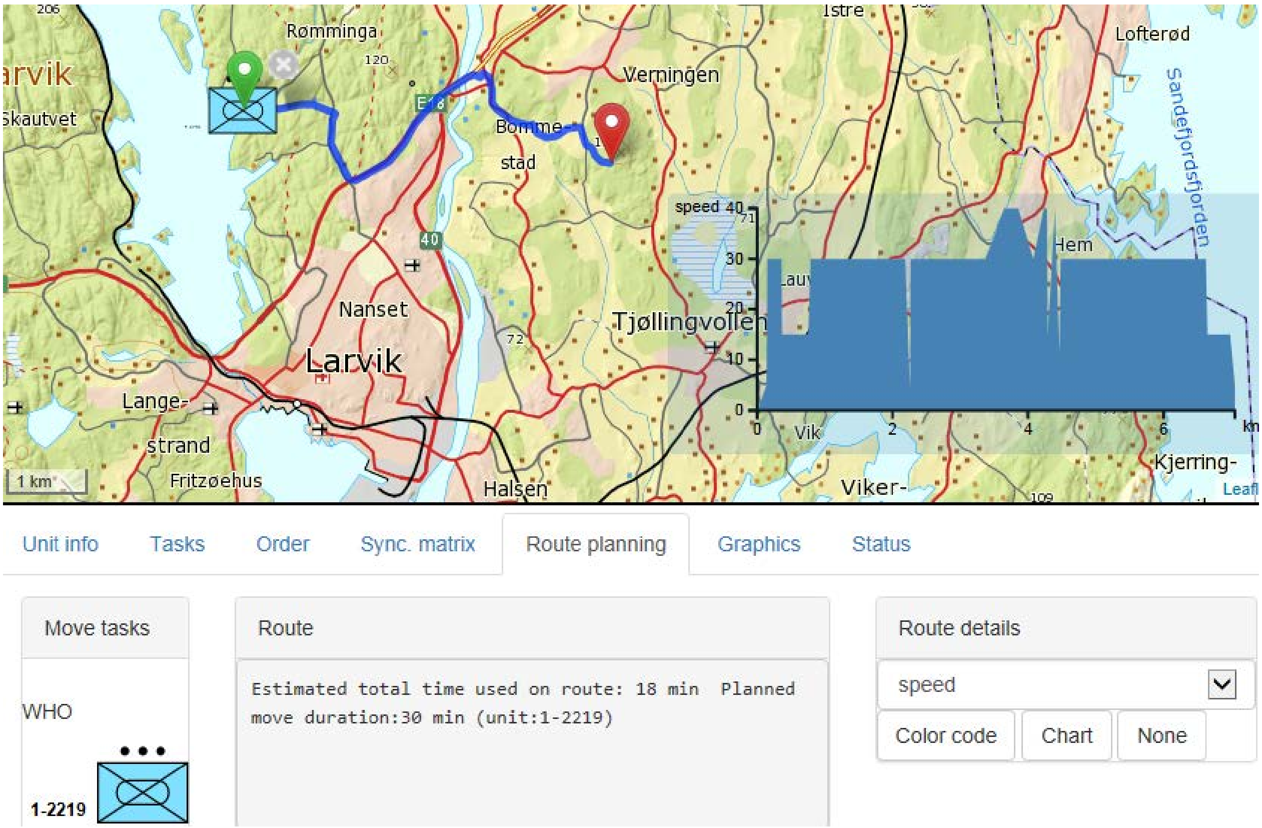

Figure 3: Screenshot of route planning view. The route planner is used to classify segments of the planned route according to speed, and to provide feedback on estimated time compared to the time planned in the order.

dedicated status view. The information displayed here reflects the hierarchy of the order of battle; as the main interest of the OPG is the actions of the units one level below in the hierarchy. By selecting a given unit, the status view will provide an overview of its subordinate units with respect to logistics, speed and remaining combat power. The OPG is normally not interested in details for all units, but should be notified if some units have a critical status. The status view therefore displays averages for all subordinate units, as well as values for the units with maximum and minimum values. Further, as the simulation evolves, it is of relevance for the OPG to view how the unit synchronization evolves. Time estimates and routes for selected units to a given point of interest can therefore be viewed. Note that this latter feature is strictly a part of the predictive mode, and could also be used prior to simulation start-up.

Transparency and user control has been important when developing the UI, to maintain SWAP as a passive DSS and the OPG as the active decision maker. The user is not to be told what to do, but information is given to support decisions. As an example the route planning service provides one possible route and accompanying time estimate, modifying the route or the timing may be necessary based on operational preference and experience. The route planning view will then provide information on the relationship between time estimates from the route planner and the timings given in the plan, but it is up to the user to act upon this information (Fig. 3). Similarly for the status views; asynchronous units or resource deficiencies are revealed, but only when the information is required from the OPG.

5.0 SYSTEMS DEVELOPMENT PROCESS

We used the High Level Architecture (HLA) [13] for the simulation. The Distributed Simulation Engineering and Execution Process (DSEEP) IEEE 1730-2010 [14] is a standard for developing and executing distributed simulation systems. It is, basically, a traditional systems engineering process where simulation-

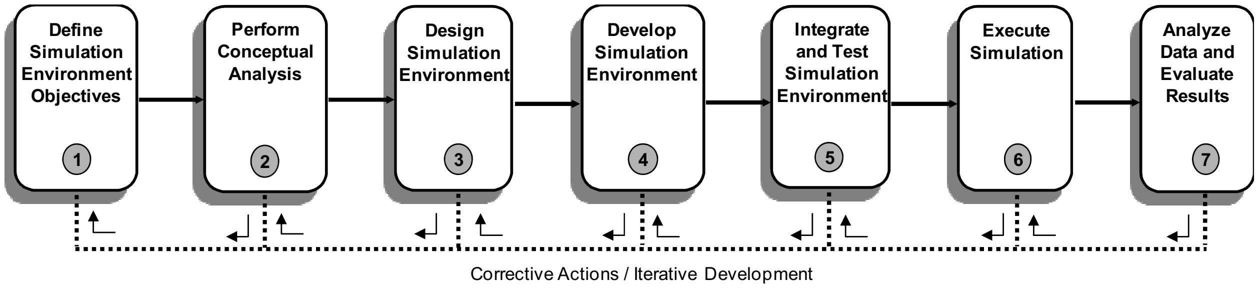

Figure 4: Distributed Simulation Engineering and Execution Process (DSEEP), main steps [14].

specific considerations are addressed, with specific simulation architectures (e.g., HLA) further detailed out in overlays to the DSEEP. Due to the fact that several components and connections were reused from earlier systems, the DSEEP was not followed in all its detail. However, development did follow the top-level structure of its seven steps; see Fig. 4. It is worth noting that the system under development included a simulation as a subsystem in a wider C2SIM context. Therefore, the perspective was broader than that covered explicitly in the DSEEP; a perspective captured in ongoing work on a C2SIM overlay to DSEEP [15].

As SWAP is a passive DSS, offering support to the decision maker, the requirements handling process focused on understanding the planning process and on capturing how users would interact with the system.

Inspired by [16], we followed a user story approach based on user stories at various levels of abstraction and elaboration and refinement. High-level user stories go under the name of "Epics" and function as placeholders for more elaborated and refined user stories (usually called, simply, "stories") developed later. A schematic of the approach is provided in Fig. 6. In line with [16], the user story approach was structured according to the C3 Taxonomy; see Fig. 5 for a high-level view. The taxonomy enables the defense community to sort capabilities into meaningful pieces to aid the definition of services at all levels. It explicitly includes, in the same picture, the operational context (Operational Context frame in the C3 Taxonomy) and the computing context (Communication and Information Systems (CIS) Capabilities frame). Incidentally, figures in our discussion are colour coded according to the C3 Taxonomy's candy color chart [17] where relevant.

We started with a definition phase (DSEEP Step 1) in which the needs of the planning group were identified. This was done through observations of the planning group and through discussions with subject matter experts. A high-level operational epic in the Business Processes layer (Fig. 5) was formulated to identify at what stage in the Norwegian PBP simulation support could be beneficial in terms of the three propositions above:

> E0: *As Brigade OPG, I can use simulation support to develop, visualize and analyze COAs according to the Norwegian PBP, in order to P1, P2 and P3.*

This epic was later elaborated and refined as described below.

A conceptual model for what the simulation system should simulate was developed (DSEEP Step 2). The system was not to be bound to a particular scenario or simulation event. Rather, the system was designed for decision support for various operational scenarios. Therefore, the conceptual model has a level of detail such that it defines the minimum requirements a simulation scenario must meet to fulfill Epic E0.

Alongside E0, the conceptual model laid the premises for the design and development of the simulation (DSEEP steps 3-4). The development was driven by user stories at the CIS Capabilities level of the C3 Taxonomy, which is the context of SWAP users and SWAP components. DSEEP prescribes steps to identify which member applications (HLA federates) are needed in the simulation, to design them and

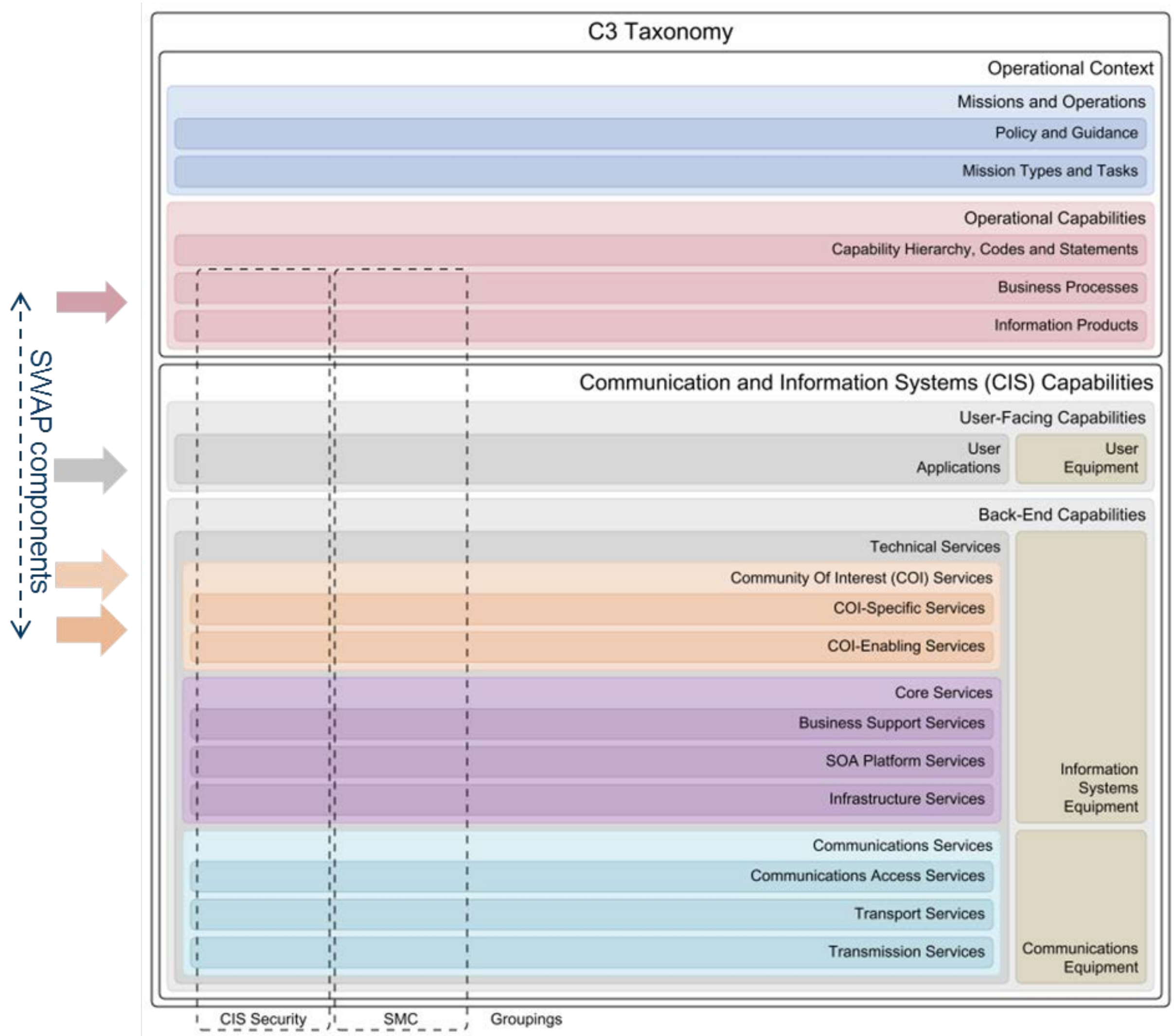

Figure 5: C3 Taxonomy – top-level view, with SWAP components indicated

to develop them. The nature of our process was somewhat different from this perspective, because our member applications were remakes of older versions of components, and because they were adaptations of a simulation framework (VR-Forces) on the one hand, and highly specialized federates (Multi-Agent System and MAESTRO Time Management) with limited interaction with other federates on the other hand.

Epic E0 was elaborated into a set of epics at the User Applications level of the C3 Taxonomy, of which a subset were to be addressed by SWAP; for example:

> E1.1: *As Brigade OPG, I can use SWAP to develop blue COAs according to the Norwegian PBP, by importing CONOPS from Joint HQ, refine CONOPS to COAs, wargame COAs by interactive simulation and use metrics from simulations to analyze COAs, in order to achieve E0.*

These epics were further detailed into stories at the User Applications level. Of particular interest to our subject matter experts was how the synchronization matrix could be formed and refined in step with the interactive simulation, and how best to visualize the execution of COAs and plan sketches.

Relevant stakeholders were identified and mapped to the relevant levels of the C3 taxonomy. At each level, the user stories acted as mandating epics for user stories at lower levels (Fig. 6).

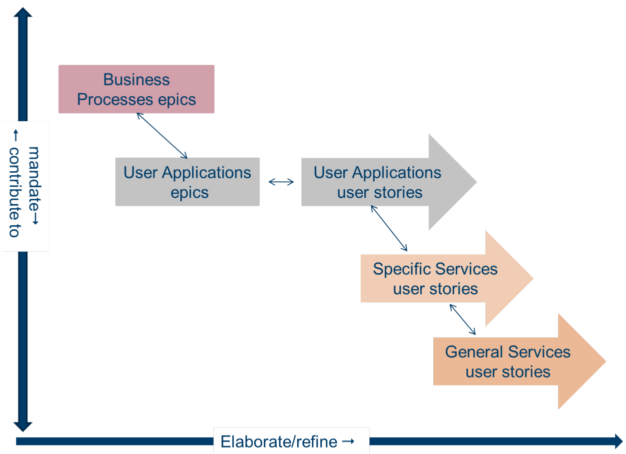

Figure 6: Implementation of user story driven approach, using the levels of the C3 Taxonomy. An overall epic from Business Processes were refined/elaborated into a set of epics for User Applications. These were in turn refined/elaborated into User Application User stories. In the following steps, user stories at one level mandated user stories at the levels below.

As we started out with a broad mapping of OPG needs, the resulting map of user stories was more extensive than the scope of SWAP. A road map towards further development was therefore achieved.

The final two steps of DSEEP, 6 and 7, are currently not finalized. We plan to use SWAP to perform simulation-supported wargaming in cooperation with an OPG during a planning process and evaluate the results of this experience.

6.0 SYSTEM OVERVIEW

The system has a hybrid architecture, in that it has a specialized architecture (HLA) residing inside a larger federated architecture. The intention is that the larger architecture should follow service-oriented architecture (SOA) principles [18]. The reference architecture in Fig. 7 is a blueprint for our systems architecture in Fig. 8. The reference architecture is a blueprint for HLA federations in a wider context of C2IS (C2SIM configuration), together with services, in an encompassing federation. (Note that we use "federation" in a general systems engineering sense at the encompassing level, while a HLA federation is a highly specialized type of federation.) A key issue in the reference architecture is the use of interoperability standards to enable standardized interfaces and flexibility in configuration. The color coding follows the levels of the C3 Taxonomy. At the reference architecture level, (service) descriptions (i.e., syntactic and semantic specifications of functionality) are the salient artifacts, while concrete systems and software implementing those descriptions are only shown in terms of place holders (without color). Interaction patterns in the reference architecture give high-level templates for how functionality specified in the descriptions may be used between components. Such patterns are currently under development elsewhere and were not used here.

SWAP was originally envisioned as a group of independent services to be connected together. By simulating their own areas of responsibility and exchanging information through standardized interfaces, they form a coherent synthetic environment in which simulations and analyses can be run. This also makes it easy to add or replace simulation components, or even use SWAP itself as a component in a larger system, in order to broaden the scope or enhance the fidelity of the simulation. To accomplish this, we have developed SWAP as a group of separate components, which communicate with each other using well-defined interfaces over standard communications protocols. The systems architecture (Fig.8) shows the actual systems and software used, as well as syntactic descriptions of their interfaces where applicable. The data exchanged and the communications protocols used has also been indicated. The standards indicated in the figure have been followed to the extent that is practically possible at this stage.

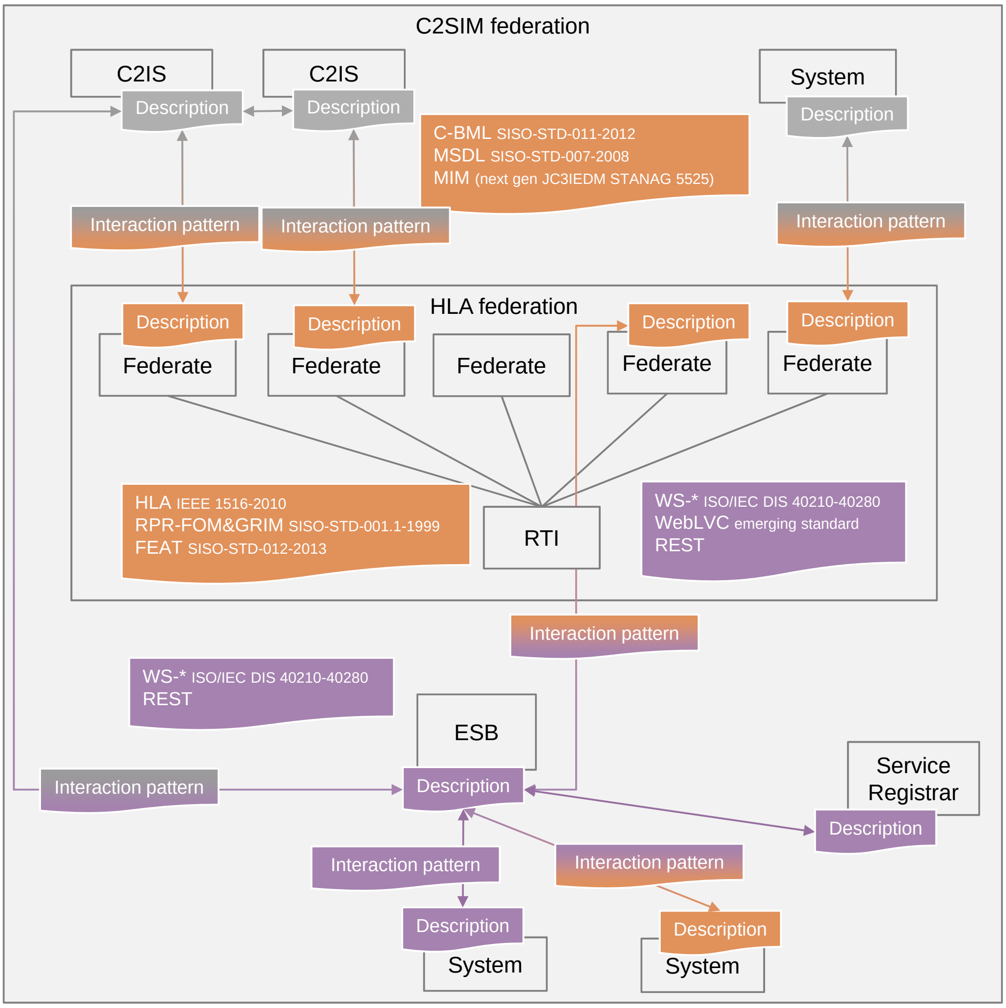

Figure 7: Reference Hybrid Architecture for C2SIM federation

The NATO Core Services Recommendation [19] states that one should use WebServices (WS*) [20], [21] binding to SOAP [22] for request-response and Publish-Subscribe (WSNotification). However, the Norwegian Defence Logistics Organisation (NDLO) has decided upon using AMQP [23] for publish-subscribe, and recently as the transport protocol as such. We therefore use AMQP. However, connectivity with NATO systems is enabled by a connection compatible with the NATO Core Services Recommendation.

The dashed rounded boxes indicate one possible configuration of system deployment, illustrating the loose coupling of the total system. The route planning service implementation, the HLA federation, the web server and NORCCIS can be hosted and deployed at different places according to, e.g., responsibility for developing, acquiring and maintaining each system. This also allows flexible deployment and accessibility of systems in different operational contexts. The Web-based User Interface (UI) is a C2IS surrogate developed because NORCCIS does not have enough flexibility to integrate easily with the other systems in our context, but also provides flexible access to the system through web technology which can run on virtually any (sufficiently powerful) device running a web browser.

Descriptions of NORCCIS and the key components of SWAP and its internal interfaces, as depicted in Fig. 8 are given in the following sections.

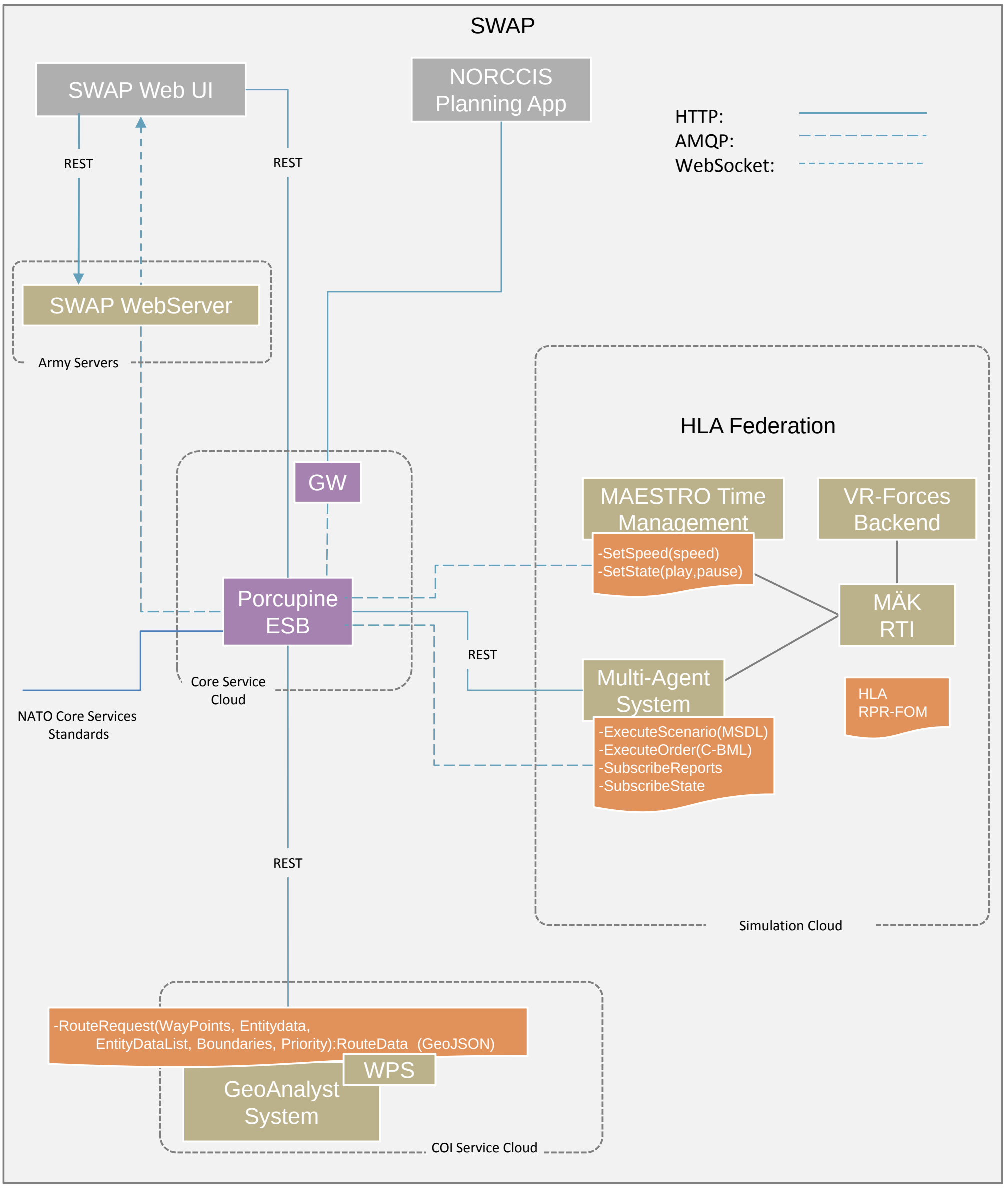

Figure 8: SWAP system architecture. Concrete systems (olive) with interfaces (orange) where applicable. Lines according to application layer protocols (HTTP, AMQP, WebSocket). Requests over lines to interfaces at provider. Responses over lines from interfaces to systems.

6.1 NORCCIS

The Norwegian Command and Control Information System (NORCCIS) is a tool used by all services of the Norwegian Armed Forces for command and control (C2), displaying the common operational picture, plans, the order of battle, and so forth. It is developed and maintained by the NDLO. SWAP has been built as a demonstrator of functionality which we envision can be included in future versions of NORCCIS. As a temporary measure, we have developed a gateway to pipeline key information into SWAP.

NORCCIS will readily export all of its data, but it does so using a proprietary format which must be decoded using libraries provided by NDLO. Of most interest to SWAP is the NORCCIS Order of Battle (ORBAT), and its tactical graphics. We have created a gateway which uses the NDLO libraries to convert the former into a JavaScript Object Notation (JSON) [24] structure closely reminiscent of the MSDL, while tactical graphics are converted into a GeoJSON [25] structure. Both are then sent to the Porcupine

Communications Server using the AMQP protocol, which is then able to distribute it further to any interested federates.

6.2 Web-based graphical user interface

The Web-based graphical user interface (UI) (Fig. 8) is the user interface to SWAP. It controls the simulation of SWAP, and is also intended to provide examples on how a C2IS could be extended to provide additional information supporting the OPG in the planning process, offering pre-analysis of plans (predictive aspect) and describing the outcome of the simulation. The UI is used to extend orders received from a C2IS, making them executable by a simulation.

The UI loads MSDL-files containing ORBAT, allowing the user to pre-analyze and edit them, before passing them on to simulation. It also displays the current situation, and lets the user give orders and tasks to units under their control.

HTML5 [26] and modern web browsers have made it possible to implement increasingly sophisticated applications directly in the browser. Although traditional desktop applications still offer richer and more advanced graphical user interfaces, web-based applications have some advantages. The main advantage is that a web application is relatively easy to deploy. You only need a web browser and a network connection. Deploying traditional applications can be cumbersome, especially if they have to run in a tightly controlled environment.

The UI is hosted by a Web Server (Fig. 8). All communications with the other SWAP components is done via the web server, while the client(s) are updated using Web Sockets.

6.3 Computer Generated Forces

The simulation of the units, their movements, and combat is done using a COTS software framework for CGF. In SWAP we needed a system for aggregate level simulations that is compatible with the HLA standard. In addition, the software needed to be extendible and configurable to our needs. VR-Forces [27] is a software product that satisfied the criteria, and was used as a framework for the CGF in SWAP.

6.4 The Multi-Agent System

The Multi-Agent System (MAS) commands and controls the behavior of the units of the CGF. It models the combat management of military leaders. The MAS allows the user to give higher level tasks to the system, as MAS interprets received tasks into lower level tasks for the simulated entities.

MAS receives ORBAT and orders from the command and control information system (C2IS) via the UI. The system can handle ORBAT and orders at the brigade level and lower levels. The MAS creates a hierarchy of intelligent agents based on the received ORBAT, with one agent for each unit in the hierarchy. This includes platoon agents, company agents, battalion agents, and at most one brigade agent, depending on the forces represented in the ORBAT. The agents represent the leaders with staff of these military units, and the MAS simulates the planning and decision making of the military leaders.

When the MAS receives an order, it decomposes the tasks through the agent hierarchy to low-level CGF commands for the leaf units of the ORBAT. The leaf units of the ORBAT and their execution of tasks are simulated by the CGF. If the lowest level in the ORBAT is companies, the lowest level agents will also be companies, and the MAS will produce company commands to the units of the CGF. The detail level of the ORBAT is allowed to vary for each branch. The behavior of the agents in the MAS is modeled using the Context-based Reasoning paradigm, as described in more detail in [28].

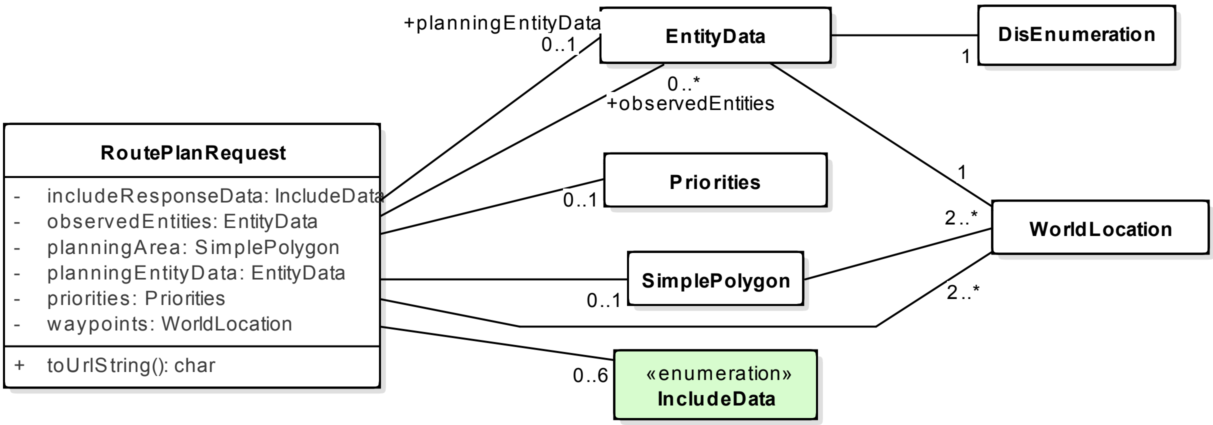

Figure 9: Description of the request to the route planning service. A request includes way points, data for the planning entity, and a bounding area for path planning. It also includes data for other relevant entities such enemies within the planning area. `IncludeData` is an enumeration for what kind of additional data to include in the response, for example speed along the route.

6.5 Route planning service

One of the challenges with simulating military operations, is for the CGF to move realistically. In SWAP, this is achieved by letting the MAS provide routes along with the tasks to the CGF. The routes are computed using a method for situation dependent path planning that was developed in a previous work [29].

A route planning capability is useful for other purposes than computing routes for the MAS. The OPG can for instance use route planning to estimate the duration of a movement task, or the CGF may want to have direct access to route planning without going through the MAS. We have therefore implemented route planning as a service. The service supports planning for various unit types and can adjust the search according to, for instance, specific areas of operation, or knowledge of enemy units.

The route planning service is implemented as a Web Processing Service (WPS) [30] in a RESTful manner, communicating over HTTP. The service uses GeoAnalystLibrary for path planning, which is a library under development in a collaboration between FFI and FOI, the Swedish Defence Research Agency.

The client interacts with the server by sending a sending an Execute operation request. The DataInputs data structure of the request contains a PlanRouteRequest, as described in Fig. 9. One of the input variables assigns priorities to the various planning categories, including time, threat level, and terrain accessibility. Details of the implemented route planning can be found in [29]. The response is a list of coordinates and corresponding metadata. The currently supported response format is GeoJSON.

The request message is sent using either HTTP GET or POST. In the case of HTTP GET the input arguments are sent as key-value pairs encoded as a string. The only required argument is `waypoints`. Any unspecified arguments will use default values, which can be obtained by sending a DescribeProcess request. A sample GET request may look like:

```
hostname/wps?
    service=wps&
    version=1.0.0&
    request=Execute&
    identifier=RoutePlanner&
    dataInputs=[
```

```
        waypoints=LineString(10.1 59.0, 10.2 59.3);
        entityData=EntityData(10.1 59.0,1 1 163 3 0 0 0,6.0,1)
    ]
```

Line endings and spaces are used for readability.

6.6 MAESTRO

The SWAP federation is tightly bound to the time regulation services found in HLA. Currently, MAS depends on HLA time advance messages as the cue to tick its agent engine. This makes it important to be able to control the passage of simulated time. The CGF is currently able to act as a time regulating federate, but the only available options are to either simulate in real-time, or to progress as quickly as possible. MAESTRO was developed in order to give more control over the progression of simulated time in an HLA federation; it can lock simulation time to nearly any rate, in addition to pausing and resuming the federation.

6.7 The Enterprise Service Bus

In order to allow all the separate components of SWAP—which may not even know about each other—to communicate, we use an Enterprise Service Bus (ESB). The ESB ensures that all messages are routed to the correct recipients. The ESB developed for SWAP is based on Apache Camel [31], and is called the Porcupine Communications Service.

Messages are distributed by using message topics as a basis for a publish/subscribe model; if a component is interested in messages pertaining to e. g. MSDL files (which contain ORBAT), it needs to subscribe to messages with the topic 'msdl' with the Porcupine. Whenever the Porcupine receives a message with a certain topic, it will push a copy of that message to every component subscribed to that topic. This is the key to ensuring modularity within SWAP: no component needs to even know of the existence of other components than the Porcupine. As long as a proper list of topics has been defined in advance, all SWAP components know what messages to expect and what messages to publish.

The Porcupine is also meant to act as a portal to other services that may be commonly required by SWAP components. At the moment, it acts as a proxy for the route planning service. When a component sends a route planning request, that request is forwarded to the route planning service without modification, and the response is also sent back unmodified. The intention is for other, future services to be routed through the Porcupine in the same way. This reduces the amount of configuration needed, as only the Porcupine needs to know where a service is hosted, letting the other components of SWAP use the service through the Porcupine. It also makes it easy to replace service providers, as long the new service provider exposes an interface compatible with the old service.

6.8 Interface between MAS and CGF

In previous work with developing demonstrators, Low-Level BML has been used for communication between a MAS and a CGF [32]. Low-level BML is a language developed for remote control of CGFs, for communication between simulations of C2 and a CGF, which in the case of SWAP is the MAS and VR-Forces. The implementation of Low-level BML is done by encoding orders using Google Protocol Buffers, and sent as part of a ApplicationSpecificRadioInteraction to the CGF. For more details on the technical aspect of the implementation, see [32].

Low-level BML has been expanded with additional tasks and reports in the work with SWAP. In particular the existing Create Entity, Create Route, and Move Along Route commands have been added. Create Entity was extended to support true aggregates. Create Route was used as is, but with a new use case in which the route is retrieved by the MAS from the route planning service. One of the extensions to Low-level BML was a *Transport by Helicopter* command, which transports an aggregated entity from its current position to a provided destination.

6.9 Interface between the user interface and MAS

The UI for SWAP needs to be able to load and modify scenarios, send orders and tasks, as well as receive status messages from the various simulated entities. To accomplish this, the UI, which is run in the user's browser, communicates with a Web Server, which in turn communicates with the Porcupine.

XML is commonly used as data exchange format between simulations and C2-systems [33]. However, it is more convenient to use JSON in web-based applications, because it is natively supported by Javascript. For this reason, the UI and the web server exchange data using JSON structures that resemble the structures used in MSDL and C-BML. This makes it straightforward to convert between XML and JSON when necessary. The Web Server currently converts the JSON scenario data into MSDL before forwarding it to the Porcupine, while orders and task data are forwarded without modification. Simulation status messages are also received in a custom JSON format which is designed to mirror object attributes as they are used in HLA.

The web server uses a REST interface to receive user requests, such as orders or scenarios. The requests are then forwarded to the Porcupine using the AMQP protocol. These messages will then be pushed out to MAS, as long as it's subscribed to the appropriate AMQP topics. Simulation status messages are sent from MAS to the Porcupine using AMQP in the same manner, and then pushed out to the web server. The Web Server then updates all connected clients by pushing the messages out on Web Sockets.

7.0 CONCLUSION

This paper describes a system for simulation support during the planning of military operations, with an emphasis on its development and architecture. Simulation-supported Wargaming for Analysis of Plans (SWAP) is a case-based decision support tool, which role is not to tell the user what to do, but to provide information to support the decision making of the user. SWAP is integrated with the NOR C2IS NORCCIS and is intended to provide a convenient way to wargame and analyze several Courses of Action (COA), which can give the user a better understanding of each proposed COA and in turn result in an improved plan.

The development process has followed the top-level structure of DSEEP, and we followed a user story approach at various levels of abstraction, from the overall purpose of SWAP down to system requirements. It has been a design goal to limit interdependency between components in order to make it easy to replace each component and to configure the system to different applications. The result is an HLA architecture residing inside a larger federated architecture which follows the principles of service-oriented architecture. The development process did support us in keeping a close connection between user needs and system requirements. The process has further, together with the service-oriented architecture, enabled us to build a system that can be easily adapted and extended. The system architecture is in line with Norwegian Defence IT strategy and is designed to be compatible with a service-oriented national defence information infrastructure. Integration into a NATO defence infrastructure is enabled by offering a connection compatible with technology leveraged in NATO Core Services recommendations.

Since NORCCIS currently does not have all functionality that is needed to perform simulation-supported wargaming, we have developed a Web-based user interface which functions as a C2 surrogate. This user interface controls the simulation in SWAP and is intended to provide examples on how the C2IS could be extended to provide additional information supporting an OPG in the planning process.

We have described the purpose and development of SWAP and its system architecture, but whether the demonstrator fulfills it purpose is currently not evalutated. We plan to demonstrate SWAP for the prospective users of the system and to evaluate its effect on the planning process through experiments.

8.0 ACKNOWLEDGEMENTS

The authors are grateful to the Operation Planning Group of the Norwegian Army Brigade and to the Norwegian Army Weapon School for helping us define user needs for simulation support to the planning process.

9.0 REFERENCES

[1] J. E. Hannay, K. Brathen, J. I. Hyndøy, On how simulations can support adaptive thinking in operations planning, in: NATO Modelling and Simulation Group Symp. M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence (MSG-133), 2015.

[2] NATO Science and Technology Organisation, Data farming in support of NATO – final report of task group MSG-088, Technical Report STO-TR-MSG-088 AC/323(MSG-088)TP/548 (2014).

[3] J. Schubert, P. Horling, Preference-based monte carlo weight assignment for multiple-criteria decision making in defense planning, in: Information Fusion (FUSION), 2014 17th International Conference on, 2014, pp. 1–8.

[4] L. Khimeche, P. de Champs, APLET aide à la plainification d'engagement tactique terrestre m&s in decision support for course of action analysis, APLET, in: NATO Modelling and Simulation Group Symposium on "Modelling and Simulation to Address NATO's New and Existing Military Requirements", RTO-MP-MSG-028, 2004.

[5] D. J. Power, Decision Support Systems: Concepts and Resources for Managers, Greenwood Publishing Group, 2002.

[6] P. Hättenschwiler, Neues anwenderfreundliches Konzept der Entscheidungsunterstützung, in: Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft, vdf Hochschulverlag AG an der ETH Zürich, 1999, pp. 189–208.

[7] A. Alstad, R. A. Løvlid, S. Bruvoll, M. N. Nielsen, Autonomous battalion simulation for training and planning integrated with a command and control information system, Tech. Rep. FFI-rapport 2013/01547, Norwegian Defence Research Establishment (FFI) (2013).

[8] R. A. Løvlid, A. Alstad, O. M. Mevassvik, N. de Reus, H. Henderson, B. van der Vecht, T. Luik, Two approaches to developing a multi-agent system for battle command simulation, in: Proc. of the 2013 Winter Simulation Conference, 2013.

[9] J. I. Hyndøy, O. M. Mevassvik, K. Bråthen, Simulation in support of course of action development, in: Proc. of the 2014 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), 2014.

[10] Forsvaret, Stabshåndbok for Hæren – Plan- og beslutningsprosessen [In Norwegian], 2015.

[11] Simulation Interoperability Standards Organization, Standard for: Military Scenario Definition Language (MSDL), http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=30830, accessed August 2012 (2008).

[12] Simulation Interoperability Standards Organization, Standard for Coalition Battle Management Language (C-BML) phase 1, http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=41767, accessed July 2014 (2014).

[13] IEEE Standards Association, 1516-2010 – IEEE Standard for modeling and simulation (M&S) High Level Architecture (HLA), http://standards.ieee.org/findstds/standard/1516-2010.html, accessed September 2012 (2010).

[14] IEEE Standards Association, 1730-2010 – IEEE recommended practice for Distributed Simulation Engineering and Execution Process (DSEEP), http://standards.ieee.org/findstds/standard/1730-2010.html, accessed February 2013 (2010).

[15] B. Gautreau, L. Khimeche, N. De Reus, K. Heffner, O. M. Mevassvik, A proposed engineering process and prototype toolset for developing C2-to-simulation interoperability solutions, in: 19th Int'l Command and Control Research and Technology Symposium (ICCRTS), 2014.

[16] J. E. Hannay, K. Bråthen, O. M. Mevassvik, Simulation architecture and service-oriented defence information infrastructures – preliminary findings, FFI-rapport 2013/01674, Norwegian Defence Research Establishment (FFI) (2013).

[17] NATO Communications and Information Agency, The C3 Taxonomy candy colour chart, http://tide.act.nato.int/em/index.php?title=Candy_Color_Chart, accessed August 2012 (2011).

[18] T. Erl, SOA principles of Service Design, Prentice Hall, 2007.

[19] NATO Consultation, Command and Control Board, Core enterprise services standards recommendations—the service oriented architecture (SOA) baseline profile, version 1.7, On Tidepedia, accessed September 2012 (2011).

[20] World Wide Web Consortium, Web Services architecture, http://www.w3.org/2002/ws/arch, accessed September 2012 (2004).

[21] World Wide Web Consortium, Web Services Description Language (wsdl) version 2.0 part 1: Core language, http://www.w3.org/TR/wsdl20, accessed September 2012 (2007).

[22] World Wide Web Consortium, SOAP version 1.2 part 0: Primer (second edition), http://www.w3.org/TR/2007/REC-soap12-part0-20070427, accessed July 2013 (2007).

[23] OASIS, OASIS Advanced Message Queuing Protocol (AMQP) version 1.0, http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf, accessed September 2015 (2012).

[24] The JSON Data Interchange Format, ECMA-404 http://json.org/ (2013).
[25] The GeoJSON Format Specification, http://geojson.org/geojson-spec.html (2008).
[26] Hypertext Markup Language (HTML), version 5, http://www.w3.org/TR/html5/.
[27] VR-Forces, http://www.mak.com/products.html.
[28] R. A. Løvlid, A. Alstad, G. Skogsrud, S. Bruvoll, O. M. Mevassvik, K. Bråthen, Modelling battle command with context-based reasoning, FFI-rapport 2013/00861, Norwegian Defence Research Establishment (FFI) (2013).
[29] S. Bruvoll, Situation dependent path planning for computer generated forces, FFI-rapport 2014/01222, Norwegian Defence Research Establishment (FFI).
[30] Open Geospatial Consortium Inc., Opengis Web Pprocessing Service, http://www.opengeospatial.org/standards/wps, accessed April 2015 (2007).
[31] Apache Camel, http://camel.apache.org/.
[32] A. Alstad, O. M. Mevassvik, Low-level BML, in: Proc. 2013 Spring Simulation Interoperability Workshop (SIW), Simulation Interoperability Standards Organization (SISO), 2013.
[33] Simulation Interoperability Standards Organization, The command and control systems – simulation systems interoperation (C2SIM) product development group (PDG) and product support group (PSG), http://www.sisostds.org/StandardsActivities/DevelopmentGroups/C2SIMPDGPSGCommandandControlSystems.aspx, accessed June 2015 (2014).