

МОДЕЛЮВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМ УПРАВЛІННЯ ТАКТИЧНОЇ ЛАНКИ НА ОСНОВІ ОНТОЛОГІЧНОГО ПІДХОДУ

В. В. Литвин, О. В. Оборська

Національний університет "Львівська політехніка"

вул. Степана Бандери, 12, м. Львів, 79000, Україна. E-mail: vasy117.lytvyn@gmail.com, oksana949@gmail.com

Розглянуто основні принципи та поняття системного аналізу, та можливість його використання при проектуванні системи керування тактичною ланкою. Внаслідок проведеного аналізу було обрано методи прийняття рішень для проєктованої системи. Математична модель представлена розв'язанням задач цілерозподілу та визначенням випадкового розподілу вогню для потоку подій. Також розглянуто інструментарій для побудови предметно-орієнтованої онтології та її управління (редактор Protege).

Ключові слова: онтологія, дуельний бій, петля Бойда.

МОДЕЛИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ТАКТИЧЕСКОГО ЗВЕНА НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА

В. В. Литвин, О. В. Оборская

Национальный университет "Львовская политехника"

ул. Степана Бандеры, 12, г. Львов, 79000, Украина. E-mail: vasy117.lytvyn@gmail.com, oksana949@gmail.com

Рассмотрены основные принципы и понятия системного анализа и возможность его использования при проектировании системы управления тактического звена. В итоге проведенного анализа было выбрано методы принятия решений для проектируемой системы. Математическая модель представлена решением задач целераспределения и определения случайного распределения огня для потока событий. Также рассмотрены инструменты для построения предметно-ориентированной онтологии и ее управления (редактор Protege).

Ключевые слова: онтология, дуэльный бой, петля Бойда.

АКТУАЛЬНІСТЬ РОБОТИ. На сьогоднішній день проблеми підвищення ефективності функціонування різних ланок людської діяльності – у техніці, економіці та багатьох інших, зокрема у військовій справі, вимагають розвитку нових методів досліджень. Імітаційне моделювання (ІМ) по суті і стало єдиним методом вирішення такого роду задач.

Метою розробки інформаційної системи (ІС) керування тактичною ланкою є аналіз основних понять і принципів імітаційного моделювання, розробка програмного забезпечення для здійснення моделювання перебігу бою на основі онтології.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. Складність і різноманіття реальних ситуацій вимагають для їх адекватного відображення в математичних моделях гнучкості та універсальності останніх. Ці властивості неминуче приходять в протиріччя з спільністю і обґрунтованістю результатів моделювання. Тому при вирішенні тих чи інших реальних завдань неминуче використання комплексів моделей, в яких «вихід» однієї моделі є «входом» для іншої і т.д. Сукупність подібних моделей може розглядатись у вигляді ієрархії (зазвичай більш низькому рівню ієрархії відповідає більш високий ступінь деталізації опису модельованих систем) або горизонтального ланцюжка, в кожному елементі якої ступінь деталізації приблизно однакова. Подібний підхід до моделювання зародився і активно розвивався в 60–70-х роках ХХ століття.

Модель процесів виявлення і класифікації об'єктів (цілей) носить стохастичний характер, залежить

від дій сил противника, видимості, ступеня радіоелектронної протидії, характеру місцевості. На основі розрахованих ймовірностей визначається кількість виявляються сил і засобів противника з реально присутніх, потім моделюється ймовірнісний процес розпізнавання/класифікації цілей, в результаті чого вони співвідносяться, наприклад, або з конкретним типом зразка одиниці військової техніки, або тільки з певним класом зразків. Потім формується підсумкова доповідь роботи засобів виявлення.

Якщо противники одноразово і одночасно приймають рішення про розподіл своїх сил «в просторі» (між плацдармами), то отримуємо гру полковника Блотто - в якій переможець на кожному з плацдармів визначається в результаті рішення відповідних рівнянь Ланчестера. Іншими словами, можна розглядати «ієрархічну» модель, в якій на верхньому рівні ієрархії гравці розподіляють свої сили між плацдармами в рамках тієї чи іншої варіації теоретико - ігрової моделі гри полковника Блотто, а на нижньому рівні результат бою на кожному з плацдармів описується тієї чи іншої варіацією моделі Ланчестера. Складність аналітичного дослідження таких ієрархічних моделей обумовлена тим, що в більшості випадків для гри полковника Блотто важко знайти аналітичне рішення.

Грою полковника Блотто називається гра двох осіб, в якій гравці одноразово, одночасно і незалежно (не знаючи вибору опонента) розподіляють свої обмежені ресурси між скінченною кількістю об'єктів (полів битв або об'єктів захисту/нападу [1], одноча-

сних конкурсів / аукціонів, груп виборців, тощо).

Позначимо через $N = \{1, \dots, n\}$ множину об'єктів, через $x = (x_1, \dots, x_n)$ – дію першого гравця, через $y = (y_1, \dots, y_n)$ – дію другого гравця, де $x_i \geq 0$ ($y_i \geq 0$) – кількість ресурсу, виділеного першим (другим) гравцем на i -й об'єкт, $i = \overline{1, n}$.

Обмеженість ресурсів відображається умовами:

$$\sum_{i \in N} x_i \leq R_x, \quad \sum_{i \in N} y_i \leq R_y.$$

Для побудови онтологічної моделі, насамперед, необхідно визначити ієрархію понять (множину C). Нехай задана множина назв відношень $V = \{v_1, v_2, \dots, v_s\}$. Тоді відношення задається як відображення із C в C , використовуючи елемент множини V : $R : C \xrightarrow{v} C$. Тобто відношення r_i – це триплет вигляду: $r_i = \langle C_{i_1}, v_{i_j}, C_{i_2} \rangle$.

Відношення ієрархії $IS - A$ та агрегації є вертикальними, їх позначають R^V . Всі інші відношення є горизонтальними і позначають R^H . Очевидно, що $R^V \cup R^H = R$, $R^V \cap R^H = \emptyset$. Вертикальні відношення задають таксономію понять предметної області. Для задання горизонтальних відношень необхідно визначити область визначення відношення (домен) та множину значень (ренг). Наприклад відношенням множини факторів (v_k), які впливають на перебіг ведення бою є: відстань між військами; характеристики ходових властивостей механізованих військ; місцевість (коефіцієнт супротиву руху, видимість цілі); сектор пошуку цілі; розподіл вогню по цілям противника; число необхідних вистрілів для знищення цілі (характеристика розсіювання, захищеність цілі, відстань). В онтологічній моделі таке відношення задається як відображення множини перерахованих факторів (домен C_i) на концепт «Перебіг ведення бою» (ренг C_j), тобто $R : C_i \xrightarrow{v_k} C_j$.

Інакше, кажучи, множина відношень R задає відображення деякої множини X (область визначення) в множину Y (множина значень), тобто $R : X \rightarrow Y$. Наведемо ряд прикладів такого відображення з військової тематики.

1. Проводиться розвідка на уточнення координат цілі. Показником ефективності може бути збільшення ймовірності враження даної цілі.

X – ціль виявлена достатньо точно, Y – операція відбулась успішно.

X – ракета попадає не далше як 500м від цілі, Y – ціль вражена.

2. Проводиться розвідка на уточнення рішення (якщо ціль вражена стрільба зупиняється, якщо ні – продовжується до використання всіх боеприпасів). Завдання – попередити зайві витрати боеприпасів на ціль, яка вже вражена. В якості показника ефектив-

ності може бути середнє число зекономлених боеприпасів (ракет).

X – ціль вражена, Y – стрільба зупиняється.

3. Проводиться напад на противника. Важливими факторами успішної операції є:

X – стрільба з літака по літаку, Y – тип цілі, дальність стрільби, швидкість цілі, швидкість стріляючого літака.

X – бомбардувальний наліт, Y – вид і розміри цілі, число літаків, що приймають участь у нападі, дальність стрільби, висота польоту, спосіб бомбометання, наявність чи відсутність радіоперешкод.

X – бомбардувальний наліт на деяку територію (площу), Y – час виконання операції, вартість затрачених ресурсів, кількість обслуговуючого персоналу, середня глибина проникнення противника на територію, що охороняється.

4. Вибір способу ведення стрільби:

X – ціль малорозмірна (одиначна ціль), Y – зосереджена стрільба.

X – ціль майданна, Y – рознесена стрільба.

5. Щоб оцінити ефективність стрільби, крім характеристик розсіювання (точність пострілів), потрібно враховувати характеристики вражаючої дії боеприпасів по цілі (конструкція, вага і руйнівна міць снаряду, конструкція і міцність цілі).

X – снаряд попав у точку з деякими координатами, Y – ціль знешкоджено.

Функції інтерпретації F задають обмеження або приймання певних значень властивостями концептів.

За допомогою процедур інтелектуального аналізу (ІА) змісту онтології отримуємо метазнання, які можна використати як знання U для управління підрозділами. Таким чином отримаємо таку структурну модель автоматизованої системи управління:

$$S = \langle O, IA, U \rangle.$$

Щоб використати процедури інтелектуального аналізу, необхідно промодельовати перебіг воєнних дій за допомогою запропонованої моделі інформаційного простору. Таке імітаційне моделювання є подальшим етапом наукових досліджень.

Основними проблемами, що виникають в інформаційних системах прийняття рішень керування тактичною ланкою є визначення систем озброєння для видів збройних сил, визначення рекомендацій щодо оптимального складу збройних та ефективний розподіл матеріальних та грошових ресурсів при розробці перспективних систем озброєння. Під час системного аналізу, було побудовано діаграму станів, діаграму діяльності, діаграму варіантів використання та діаграми класів, що служать для представлення статичної структури моделі системи.

Головне призначення діаграми станів – є описати можливі послідовності станів і переходів, які в сукупності характеризують поведінку елемента моделі протягом його життєвого циклу. Найчастіше діагра-

ми станів використовуються для опису поведінки окремих екземплярів класів (об'єктів), але вони також можуть бути застосовані для специфікації функціональності інших компонентів моделей, таких як варіанти використання, актори, підсистеми, операції та методи.

Діаграма станів по суті є графом спеціального вигляду, який представляє певний автомат. Вершинами цього графа є стани і деякі інші типи елементів автомата (псевдостани). Дуги графа служать для позначення переходів зі стану в стан. Для розуміння семантики конкретної діаграми станів необхідно представляти не тільки особливості поведінки моделюваної сутності, але і знати загальні відомості з теорії автоматів.

При моделюванні поведінки проекрованої або аналізованої системи виникає необхідність не тільки представити процес зміни її станів, але і деталізувати особливості алгоритмічної і логічної реалізації виконуваних системою операцій. Для моделювання процесу виконання операцій в мові UML використовуються так звані діаграми діяльності. Застосовується в них графічна нотація багато в чому схожа на нотацію діаграми станів, оскільки на діаграмах діяльності також присутні позначення станів і переходів. Відмінність полягає в семантиці станів, які використовуються для подання не діяльностей, а дій, а також у відсутності на переходах сигнатури подій. Кожен стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї операції в попередньому стані. Графічно діаграма діяльності представляється у формі графа діяльності, вершинами якого є стани дії, а дугами – переходи від одного стану дії до іншого.

У контексті мови UML діяльність (activity) є деякою сукупністю окремих обчислень, виконуваних автоматом. При цьому окремі елементарні обчислення можуть призводити до деякого результату або дії (action). На діаграмі діяльності відображається логіка або послідовність переходу від однієї діяльності до іншої, при цьому увага фіксується на результаті діяльності. Сам же результат може призвести до зміни стану системи або поверненню деякого значення.

На (рис. 1) зображено діаграму активності для розробленої системи. З діаграми можна побачити, що в процесі взаємодії системи з користувачем існуватиме три доріжки:

- користувач;
- інтелектуальна система;
- онтологія.

Спочатку користувач виконує такі дії:

- вводить дані про карту (розміри поля бою, рельєф, замаскованість, прохідність);
- вводить дані про бойову техніку (кількість бойових одиниць, їх тип та координати);

– задає точність моделювання (кількість ітерацій, крок моделювання).

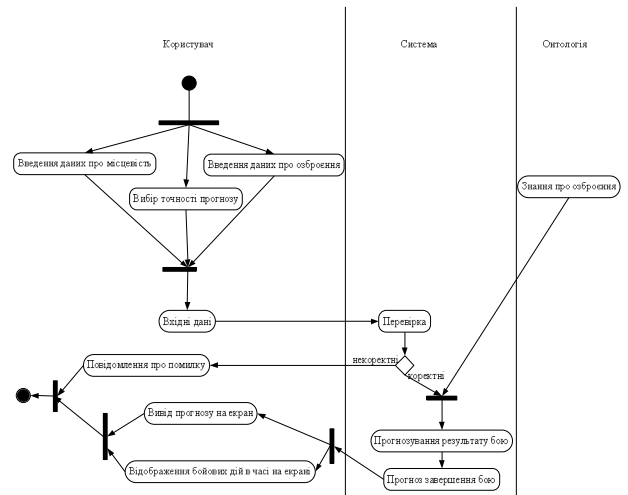


Рисунок 1 – Діаграма діяльності

У результаті цих дій формуються вхідні дані. Наступною дією система перевіряє вхідні дані. Якщо вони неправильні виводиться повідомлення про помилку, інакше на основі вхідних даних і знань в онтології робиться прогноз про перебіг бою та відбувається моделювання бойових дій між «червоними» та «синіми», результати виводяться на екран. Після цього система переходить у кінцевий стан.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів може відображати, зокрема, різні взаємозв'язки між окремою сутністю предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відношень. На цій діаграмі не вказується інформація про тимчасові аспекти функціонування системи. З цієї точки зору діаграма класів є подальшим розвитком концептуальної моделі проекрованої системи.

Діаграма класів є деякий граф, вершинами якого є елементи типу "класифікатор", які зв'язані різними типами структурних відношень. Слід відмітити, що діаграма класів може також містити інтерфейси, пакети, відношення й навіть окремі екземпляри, такі як об'єкти і зв'язки. Коли говорять про цю діаграму, мають на увазі статичну структурну модель проекрованої системи. Тому діаграму класів прийнято вважати графічним поданням таких структурних взаємозв'язків логічної моделі системи, які не залежать або інваріантні в часі.

Окрім структури класів на відповідній діаграмі, указуються різні відношення між класами. При цьому сукупність типів таких відношень фіксована в мові UML і зумовлена семантикою цих типів відношень.

На діаграмах варіантів використання відображається взаємодія між варіантами використання, що

представляють функції системи, і діючими особами, що представляють людей або системи, які отримують або передають інформацію в дану систему. З діаграм варіантів використання можна отримати досить багато інформації про систему. Цей тип діаграм описує загальну функціональність системи. Користувачі, менеджери проектів, аналітики, розробники, фахівці з контролю якості та всі, кого цікавить система в цілому, можуть, вивчаючи діаграми варіантів використання, зрозуміти, що система повинна робити. До базових елементів діаграми відносяться варіант використання, актор і інтерфейс.

Варіант використання застосовується для специфікації загальних особливостей поведінки системи або іншої сутності без розгляду її внутрішньої структури (наприклад, оформлення замовлення на купівлю товару, отримання інформації про кредитоспроможність клієнта, відображення графічної форми на екрані монітора).

Актор – це зовнішня сутність, що моделюється в системі, яка взаємодіє і використовує її функціональні можливості для вирішення певних завдань. При цьому актори служать для позначення узгодженої множини ролей, які можуть відігравати користувачі в процесі взаємодії з проєктованою системою. Ім'я актора має бути достатньо інформативним з точки зору семантики, наприклад клієнт банку, продавець магазину, пасажир авіарейсу, водій автомобіля, стільниковий телефон.

Так як у загальному випадку актор завжди знаходиться поза системою, його внутрішня структура ніяк не визначається. Для актора має значення тільки його зовнішнє уявлення, тобто то, як він сприймається з боку системи. Актори взаємодіють з системою за допомогою передачі і прийому повідомлень від варіантів використання. Повідомлення являє собою запит актором сервісу від системи і отримання цього сервісу. Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами і варіантами використання або класами. Крім цього, з акторами можуть бути пов'язані інтерфейси, які визначають, яким чином інші елементи моделі взаємодіють з цими акторами.

Інтерфейс служить для специфікації параметрів моделі, які видимі ззовні без вказівки їх внутрішньої структури. У діаграмах варіантів використання інтерфейси визначають сукупність операцій, які забезпечують необхідний набір сервісів або функціональності для акторів. Інтерфейси не можуть містити ні атрибутів, ні станів, ні направлених асоціацій. Вони містять лише операції без вказівки особливостей їх реалізації. Формально інтерфейс еквівалентний абстрактному класу без атрибутів і методів з наявністю тільки абстрактних операцій [2].

В ході виконання роботи було створено програму, яка взаємодіючи з онтологією здійснює прогноз результатів бою. Нижче наведено перелік, призна-

чення та опис основних інтерфейсів програми.

– CellInterface – призначений для представлення елементарної ділянки на полі бою. Класи, що реалізують цей інтерфейс інкапсулюють дані про видимість, прохідність, висоту над рівнем моря цієї ділянки.

– MashineInterface – призначений для опису бойової машини. Класи, що реалізують цей інтерфейс інкапсулюють дані про бойову машину (швидкострільність, тип гармати, потужність двигуна і т.д.), боездатність, поточну позицію.

– FieldInterface – призначений для опису поля бою. Класи, що реалізують цей інтерфейс інкапсулюють дані елементарні ділянки та розміри поля бою.

– ModelInterface – описує тип моделі, яка використовується в СППР.

– WeaponInterface – описує зброю, яка може розміщуватися на бойовій машині. Класи, що реалізують цей інтерфейс інкапсулюють дані про розсіювання по x і y, радіус стрільби та швидкострільність.

– WayInterfallInterface – описує інтервал шляху руху танку.

– ShotInformationInterface – призначений для опису інформації про результати пострілу

– GranateInterface – призначений для опису снарядів. Класи, що реалізують цей інтерфейс інкапсулюють дані про радіус ураження.

Оцінка функціональності програм для побудови онтологій залежить від:

- практичної задачі (цілі розробника);
- області знань, в рамках якої будується онтологія;
- онтології що розробляється.

Для онтології важливим критерієм є можливість перенесення її на інші платформи, можливість перевести її на інші формальні мови. Так, засіб створення онтологій Protege – вільно розповсюджувана локальна Java програма на базі Windows, дозволяє виконувати експорт в RDF, RDFS, XML, HTML, OWL, Clips, N3, TURTLE. Protege була що розроблена групою медичної інформатики Стенфордського університету. Програма призначається для побудови (створення, редагування та перегляду) онтологій моделей прикладної області. Її ціль – допомогти розробникам програмного забезпечення у створенні і підтримці явних моделей предметної області і у включенні цих моделей безпосередньо в код програми. Protege складається з трьох головних частин, що повинні використовуватися по черзі. Спочатку йде редактор онтологій, що дозволяє проектувати онтології розвертаючи ієрархічну структуру і включаючи абстрактні або конкретні класи і слоти. Ґрунтуючись на сформованій онтології, Protege здатний генерувати інструмент придбання знань для введення екземплярів онтології. Остання частина програми - інтерпретатор схем, який дозволяє робити екземп-

ляри для класів і підкласів. Інструмент має повний графічний інтерфейс, що є дуже зручним для використання недосвідченими користувачами [3].

Нехай необхідно дослідити закономірності результату бою танкової роти (батальйону) при наступі на протитанковий укріплений район.

Розглядається наступне тактичне завдання. Танкова рота «синіх», що складається з m танків, повинна прорвати протитанковий укріплений район «червоних». Цей район обороняє n танків «червоних», які замасковані і знаходяться в спеціально створених укриттях. Рота повинна наступати в заданому бойовому порядку в смугу шириною a метрів і глибиною c метрів. Загальний напрямок руху роти визначається взаємним розташуванням на місцевості вихідної позиції «червоних» і «синіх». Напрямок руху для кожного танка «синіх» визначається сукупністю орієнтирів, якими є добре розрізнювані місцеві предмети.

Бій починається в деякий заданий час T_0 і триває до того моменту, коли сили однієї зі сторін, що борються стануть небоездатними. Кінцем бою можна також вважати той момент $T_1 > T_0$, в який буде виконана бойове завдання або втрати однієї зі сторін перевищать допустимий відносний рівень. Головне вікно програми з картою та взаємним розміщенням бойових машин зображено на (рис. 2).

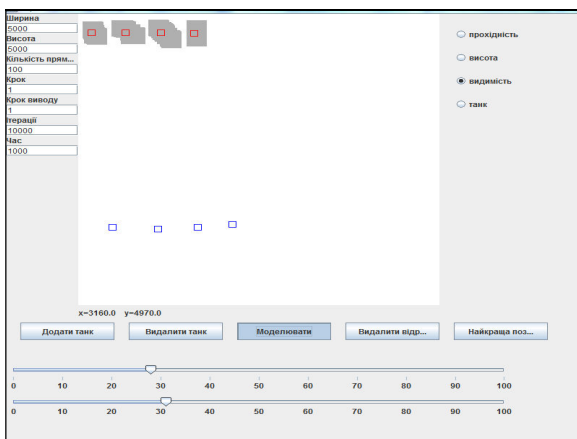


Рисунок 2 – Головне вікно програми

Бій розпочнеться приблизно в момент $T_0 + 65c$, оскільки бойові машини знаходяться на відстані, що перевищує радіус пострілу, то $65c$ потрібно, щоб ця відстань стала допустимою для початку атаки. Бій закінчується в момент $T_1 = T_0 + 180c$. Починаючи з цього моменту, графік ймовірностей стає паралельним, до осі x , отже, це кінець бою. Обчислимо ймовірності виграшу кожної із сторін:

$$P(\text{виграш} = \text{"сині"}) = \frac{0,2}{0,4 + 0,2} = 0,33,$$

$$P(\text{виграш} = \text{"червоні"}) = \frac{0,2}{0,4 + 0,2} = 0,66.$$

Більшу ймовірність виграшу червоних можна пояснити кращою захищеністю їхніх військ, оскільки за умовою задачі «червоні» замасковані і знаходяться в спеціально створених укриттях.

Розмістимо тепер на карті дві однакові бойові машини («червону» і «синю»), так щоб відстань між ними не перевищувала радіус пострілу і позначимо добре замасковану ділянку. Цю ділянку виберемо так, щоб жодна бойова машина не знаходилася в ній (рис. 3).

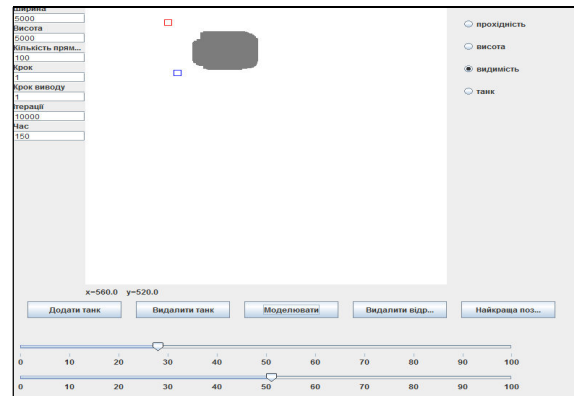


Рисунок 3 – Дуельний бій

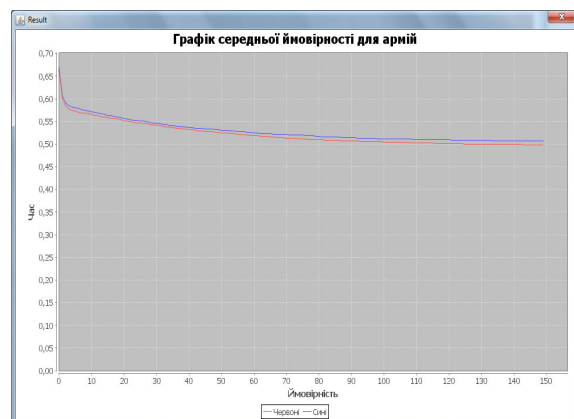


Рисунок 4 – Результати дуельного бою

На (рис. 4) бачимо динаміку бою. Оскільки бойові машини однакові, то ймовірності їх виграшу теж однакові. Незначну перевагу синіх можна пояснити похибкою моделювання.

Знайдемо найкращу позицію для атаки (рис. 5). Найкраща позиція зображена заштрихованою ділянкою.

Тепер переставимо «синю» бойову машину в цю позицію і знову спрогнозуємо результати. Тепер ймовірність виграшу синіх значно зросла (рис. 6).

З наведених прикладів можна зробити висновок, що програма працює правильно у різноманітних ситуаціях.

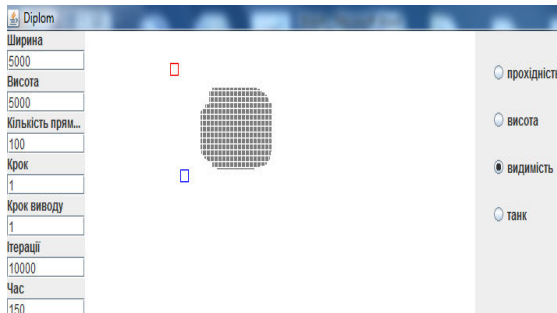


Рисунок 5 – Найкраща позиція

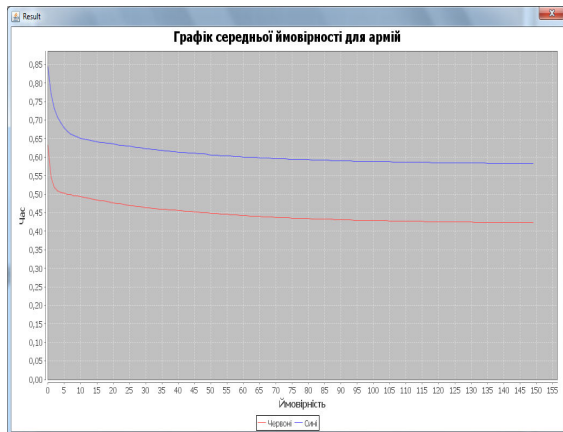


Рисунок 6 – Прогноз динаміки бою

ВИСНОВКИ. Внаслідок проведеного аналізу було обрано методи прийняття рішень для проектованої системи. Математична модель представлена розв'язанням задач цілерозподілу та визначенням випадкового розподілу вогню для потоку подій.

MODELING OF AUTOMATED CONTROL SYSTEMS TACTICAL LEVEL ONTOLOGY-BASED

V. Lytvyn, O. Oborska

Lviv Polytechnic National University

vul. Stepana Bandery, 12, Lviv, 79000, Ukraine. E-mail: vasy117.lytvyn@gmail.com, oksana949@gmail.com

The basic principles and concepts of systems analysis, and the possibility of its use in the design of control systems tactical level. As a result of the analysis was selected decision-making methods for the projected system. The mathematical model is presented for solving the problem and the definition distribution purposes random distribution of heat to the flow of events. Also the tools to build domain-specific ontologies and its management (Editor Protege).

Key words: ontology, duel fight, Boyd loop.

REFERENCES

1. Gross, O., Wagner, R. (1950) "A continuous colonel blotto game", *RAND corporation RM*, vol. 408, 13 p.
2. Litvin, V., Shakhovskaya, N. (2011) *Proektuvannya Informatsiynih sistem: navchalnyi posibnik* [Designing information systems: a tutorial], Magnolia – 2006, Lviv, Ukraine.
3. Gruber, T. (1993) "A translation approach to portable ontologies", *Knowledge Acquisition*, no. 5 (2), pp. 199–220.

Також розглянуто інструментарій для побудови і управління онтологіями.

Така система реалізована для механізованих і танкових підрозділів на основі онтологічної моделі, яка побудована в середовищі Protege. Перевагою такого підходу є те, що використання бази знань дає змогу реалізувати логічне виведення, а отже будувати план дій (другого і третього кроку петлі Бойда) [4]. Окрім того, онтології стали світовим стандартом подання знань у форматі OWL, а отже наш модуль легко може інтегруватись у будь-яку систему підтримки прийняття рішень.

ЛІТЕРАТУРА

1. Gross O., Wagner R. A continuous colonel blotto game // *RAND corporation RM*. – 1950. – Vol. 408. – 13 p.
2. Проектування інформаційних систем: навчальний посібник / В.В. Литвин, Н.Б. Шаховська. – Львів: «Магнолія–2006», 2011. – 380 с.
3. Gruber T. A translation approach to portable ontologies // *Knowledge Acquisition*. – 1993. – №. 5 (2). – PP. 199–220.
4. Онтология военных технологий на основе концептуальных карт / Ивлев А.А., Артеменко В.Б. // *Электронный научный журнал «Исследовано в России»*. – 2011 [Электронный ресурс]. – Режим доступа: [http:// zhurnal.ape.relam.ru/articles/2001/023.pdf](http://zhurnal.ape.relam.ru/articles/2001/023.pdf).

Стаття надійшла 15.10.2014.