

Віктор Євгенович Бобильов (канд. військ. наук, с.н.с.)

Юрій Миколайович Зінченко (канд. техн. наук)

Сергій Миколайович Кононенко

Національний університет оборони України імені Івана Черняхівського, Київ, Україна

ПРИНЦИПИ МОДЕЛЮВАННЯ

Створення інтерактивних віртуальних світів в комп'ютерних іграх, інтернет-просторах і моделях військового призначення останнім часом стало добре налагодженим процесом, заснованим на досвіді окремих проектувальників і програмістів включно. Однак з появою все більшої кількості таких моделей та ігор формуються принципи (основоположні, основні та інші) їх реалізації, які характерні для всіх проектів подібного роду. У статті описуються основні принципи створення програмних продуктів, які отримані шляхом аналізу розвитку

***Ключові слова:** комп'ютерна гра; модель; програмний продукт; принцип; правило; вихідні данні; моделювання.*

Вступ

Принципи моделювання, представлені в цій статті були отримані у результаті аналізу багаторічного досвіду створення комп'ютерних моделей військового призначення у відповідних структурах міністерств оборони різних країн світу. Комп'ютерні моделі військового призначення дуже схожі на комп'ютерні ігри і в деяких випадках є комп'ютерними іграми майбутнього. Тому у принципів, що будуть описані нижче, може бути поточна і перспективна застосовність для розробників комп'ютерних програм та ігор.

Постановка проблеми. У збройних силах провідних країн світу модель визначають як "абстракцію, яка представляє систему або поведінку системи в рамках певних обмежень". Моделювання – "система, яка описується моделлю з метою її вивчення, аналізу або передбачення її функціонування". У цьому контексті важливо вказати, що й самі моделі й процес моделювання будуть являти собою об'єкти і події віртуального світу близько до реальності, хоча рівень їх деталізації може значно змінюватися. Моделюватися можуть, наприклад, окремі зразки озброєння та військової техніки, їх структурні елементи (зброя, сенсори та ін.), фізика руху і балістичні характеристики окремих боєприпасів. З іншого боку можна моделювати сотні одиниць техніки і озброєння з достатньою точністю, які представлені єдиним оперативно-тактичним символом (знаком), а їх бойові можливості описувати однієї змінною; ландшафт місцевості, що покриває сотні й тисячі кілометрів і співвідношення сил протидіючих сторін. В обох випадках точність системи менш важлива, ніж послідовність взаємодії між об'єктами всередині її. Послідовність взаємодії дозволяє користувачеві регулювати його уявлення про реальний світ і входити в нього на відповідному рівні.

Моделі істотно розрізняються, однак по суті описують найбільш важливі особливості реального світу і переводять їх у віртуальний світ.

У всіх випадках цей процес відбувається вдало, якщо розробники дотримуються деяких основних принципів. Не можна стверджувати, що кожен принцип, є набором корисних рекомендацій, які можуть служити неоціненними рекомендаціями для розвитку моделювання в цілому, а також для створення моделей, ігор і віртуальних світів. Нижче розглянемо ці принципи.

Аналіз останніх досліджень і публікацій.

У галузі моделювання існує достатньо багато результатів досліджень з питань створення програмних продуктів різноманітного призначення [1, 2, 3, 6, 7]. Всі вони опубліковані у виданнях, що займаються популяризацією досягнень сучасної науки у галузі інформаційних та комп'ютерних технологій. Багато досліджень проведено і у напрямку визначення положень та основних принципів створення сучасних програмних продуктів [7, 8, 9]. Але всі ці дослідження проведені у межах своєї конкретної предметної області (наприклад: медицини, економіки і т. ін.). У військовій сфері такі дослідження проводились також [4, 5, 6], але ніде вони не були чітко систематизовані.

Тому автори зробили спробу, спираючись на свій досвід та досвід своїх колег у галузі розробки програмних продуктів військового призначення, визначити принципи моделювання, які охоплюють основні підходи до створення програмних продуктів військового призначення та систематизувати їх.

Метою статті є визначення основних принципів моделювання при створенні програмних продуктів військового призначення (моделювання бойових дій військ).

Виклад основного матеріалу дослідження

Існує принцип, який є **основоположним**. Це керівний принцип у процесі створення моделі, комп'ютерної гри або віртуального світу військового призначення. Його можна сформулювати наступним чином.

У моделі немає вродженої власної цінності. Цінність моделі повністю базується на величині ступеня вирішення якоїсь реально існуючої проблеми.

Цей основоположний принцип легко приймається, але важко йому слідувати.

У військовій сфері основоположний принцип визначає спрямованість моделі, тобто для чого модель необхідна: навчання, аналізу або вивчення поведінки системи військового призначення. Він диктує рівень роботи, яку необхідно виконати, щоб вирішити проблему, визначити додаткові опції, які можуть бути додані у кожен модель, обсяг даних, представлений в інтерфейсі й інші особливості. Основоположний принцип змушує нас створювати моделі, що описують поточні та майбутні проблеми, з якими зіткнуться користувачі цих моделей.

Основоположний принцип базується на сукупності правил, описаних нижче, які визначають логіку процесу створення моделі (моделювання).

Правило № 1: моделі не можуть бути універсальними, вони розробляються у певних цілях.

У кожного користувача є відповідний набір вимог до моделі. Іноді ці вимоги близькі до вимог користувачів, які були пред'явлені до моделі раніше. Іноді деякий сегмент вимог розходиться з існуючими на ту чи іншу модель. Тому модель, яка вчора була прекрасним рішенням задачі (проблеми), може повністю не відповідати сьогодняшнім вимогам рішення такої ж задачі (проблеми). Був час, коли система рівнянь Ланчестера (вперше представлені в 1916 році) була відмінним рішенням для моделювання процесу ведення прямого вогню. Але з кожним разом кількісні показники, отримані за допомогою рішення цих рівнянь, стають все менш близькими до реальності. Сьогодні необхідні нові методи для вирішення тих же завдань.

Правило № 2: модель вже вирішеної проблеми ніколи не буде використовуватися.

У Міністерстві оборони є фонд алгоритмів і програм, що описує майже всі існуючі моделі, що належать Міністерству оборони. Цей фонд включає в себе велику кількість моделей і задач, багато з яких були рішеннями певної задачі (проблеми). Тому вони звичайно обмежені рішенням певного завдання (проблеми). Як тільки та чи інша задача (проблема) вирішена, модель більше не має широкого використання. Те ж саме відбувається, якщо задача (проблема) виникає в нових умовах або до неї пред'являються нові вимоги. Моделі, які не відповідають цим новим вимогам також потраплять в далекі кути сховищ, щоб ніколи не бути витягнутими звідти.

Військові ігри мають точно таку ж долю. Багато військових ігор можуть бути використані для розіграшу бойових дій, але тільки деякі з них задовольняють потребам користувача.

Правило № 3: набуті знання про моделі дозволяють краще навчати процесу моделювання.

Люди, які знають все про старі моделі, ігри або методи моделювання, є чудовими джерелами ідей,

уроків і досвіду створення програмних продуктів отриманих з минулого. Однак ці знання про минуле повинні бути об'єднані з розумінням основних принципів для створення нового.

Крім основоположного принципу існує ще ряд основних принципів моделювання, які будуть розглянуті нижче.

Основні принципи моделювання

Багаторічний досвід створення програмних продуктів, постійних контактів та обміну досвідом з іншими розробниками дозволив визначити десять принципів, які, на нашу думку, дозволяють створювати вдалі програмні продукти. Ці принципи побудовані на певних правилах і рекомендаціях, які дозволяють краще зрозуміти їх, що, у підсумку, дозволить створювати великі програмні комплекси.

1. Максимально спрощуйте програмний продукт.

Створюючи модель, комп'ютерну гру або комп'ютерну програму розробники можуть врахувати набагато більше деталей для опису системи військового призначення, ніж це дійсно необхідно, щоб змоделювати її якомога ближче до реальності.

Розробники програмного продукту повинні бути обмежені потребами користувача (замовника). Додаткові ідеї повинні бути збережені для подальшого використання при створенні майбутніх програмних продуктів. Якщо у створюваній програмний продукт будуть включені всі ідеї щодо його створення, то кінцевий продукт буде роздутою і заплутаною сумішшю ідей, які логічно не пов'язані або пов'язані дуже тонким логічним зв'язком. Моделі військового призначення та комп'ютерні військові ігри зосереджуються на моделюванні певних військових місій та здійснюють їх моделювання досить ефективно.

Приблизно в 1320 році сер Вільям Оккама підбив підсумок потреби в простоті, що відомо як бритва Оккама – “*essentia non sunt multiplicanda praeter necessitatem*”, що в перекладі звучить так: “гіпотези не повинні бути складними без необхідності”. Доктор Роберт Шен нон [7], основоположник теорії дискретного моделювання, заявив, що: “Тенденція полягає в тому, що майже завжди доводиться моделювати занадто багато деталей реального світу, а не занадто мало. Таким чином потрібно завжди проектувати модель навколо питань, на які вона відповідає, а не відповідає реальній системі у точності”.

2. Використовуйте досвід минулих розробок програмних продуктів.

Вдалі програмні продукти минулого були розроблені розумними і енергійними людьми, які працювали з кращими доступними інструментами програмування того часу. Вони знайшли рішення, які застосовані в будь-якому доступному комп'ютері і при цьому проявили значну винахідливість і хитрощі в розробці, щоб досягти цієї мети. Легко подивитися назад і посміхнутися тим примітивним продуктам. Але в кожному з них укладені золоті самородки, які повинні бути здобуті для створення нових програмних продуктів.

Застарілі програмні продукти, які використовуються у збройних силах, наповнені хорошими ідеями, які можуть бути знову використані при створенні нових програмних продуктів. Компактні вирішення складних проблем включені в кожен існуючий алгоритм.

Підготовлені розробники, які вивчають ці рішення, весь час уражаються тим, як складний віртуальний світ міг моделюватися на старих комп'ютерах. Креативність, яка народжена обмеженими ресурсами, здатна до досягнення нових більш досконалих результатів, що здається неможливим погано підготовленим розробникам.

Звичайно, ці старі програмні продукти дають уроки про те, як у майбутньому не зробити те ж саме. Ті ж самі лопушки, в які потрапив ваш попередник за десятиліття до цього, чекають і нового розробника.

3. Створіть концептуальну модель програмного продукту.

Енергійні і талановиті програмісти завжди прагнуть розпочати програмування негайно. Така чудова якість має бути направлена і використана на реалізацію дуже складного процесу - створення концептуальної моделі, яка буде служити проектом для створення програмного продукту. Це частина процесу проектування, у ході якого робиться спроба описати особливості реального світу, який буде представлений в програмному забезпеченні.

Концептуальне моделювання складається з відбору об'єктів, ознак, подій і взаємодій між ними, які сформують програмний продукт. До початку програмування, розробники визначають ті об'єкти, ознаки та події, які взаємодіють між собою у ході процесу функціонування системи в реальному житті, щоб сформувати повноцінний і ефективний програмний продукт. Створення віртуального світу передбачає незліченну кількість комбінацій і особливостей його функціонування. Деякі з них є необхідними, деякі інертними, а деякі фатальними. Робоча концептуальна модель визначить віртуальний світ, який працює ефективно, повноцінно і послідовно в процесі свого функціонування. Проектувальники можуть експериментувати з новими ідеями і простежити вплив концептуальних положень на інші алгоритми, які описують поведінку системи. Постійне експериментування досягає результату, який є найкращим з усіх отриманих, що дозволяє скоротити час на розробку програмного продукту.

4. Побудуйте прототип програмного продукту.

Однією з причин, по якій розробники відмовляються від створення концептуальної моделі, є надзвичайна трудність розумового припущення і визначення всього віртуального світу та інфраструктури, яка підтримує його. В цьому випадку завжди виникають питання і припущення, які не можуть бути оцінені без робочого прототипу програмного продукту. Прототип повинен бути написаний, щоб дослідити найбільш складні моменти функціонування концептуальної моделі. Немає ніякої потреби в прототипі, який був би схожий на кінцевий програмний продукт. Прототип повинен

направити програмістів по правильному шляху вирішення проблеми, наштовхнути їх на ідеї, варіанти й інструменти, які дозволять знайти краще рішення проблеми.

У прототипу та ж сама мета, що і у концептуальної моделі - роз'яснити структуру, алгоритми функціонування і можливості кінцевого програмного продукту. Створення прототипу обмежується часом на його реалізацію і фінансами, виділеними на цей процес. Обидві ці вимоги важливі, але не для кінцевого програмного продукту. Прототип - це інструмент, який допомагає створити кращий програмний продукт, а не замінює його або виправдовує неможливість його створення. Не треба чекати, що при розробці прототипу будуть вирішені всі проблеми, будуть отримані відповіді на всі питання і виправлені всі помилки, з якими розробники зіткнуться при створенні програмного продукту. Цей принцип просто допомагає скоротити кількість і серйозність майбутніх проблем програмного продукту.

Нарешті, можна процитувати Білла Джоя з Sun Microsystems: "Великі вдалі системи виходять з маленьких вдалих систем".

5. Врахуйте актуальні запити користувача.

Програмісти, які створюють комерційні комп'ютерні ігри, краще враховують актуальні запити користувача, ніж програмісти, які створюють програмні продукти військового призначення. Бажання створити супер модель, якою будуть захоплюватися керівники проекту, іноді призводить до створення програмного продукту, який гарний тільки для вирішення неправильної проблеми. Є багато систем моделювання, які ніколи не використовуються, тому що вони вирішили проблему, якої немає.

Група розробників програмного продукту повинна знаходитися у контакті з майбутнім користувачем і зрозуміти, що він хоче отримати на виході. В яких ситуаціях користувач використовує програмний продукт, що він повинен включати в себе? Що реально спрощує їх роботу з програмним продуктом? Що змушує їх рекомендувати програмний продукт іншим? Що не влаштовує їх у програмному продукті? Що вони намагаються змоделювати, але цьому заважають обмеження моделі? Що не потрібно користувачеві у поточному наборі інструментів програмного продукту?

Новий програмний продукт повинен враховувати все краще, що було успішно реалізовано в старих продуктах, але при цьому подолати їх обмеження. Досягнення успіху не передбачає дублювання іншого програмного продукту (хоча іноді - це рішення) і вимагає, щоб був досягнутий високий рівень задоволення запитів користувачів.

Легко потрапити в пастку створення програмного продукту, який хоче розробник, а не який хоче користувач.

6. Створюйте програмні продукти на основі доступних вихідних даних

Програмні продукти військового призначення, що призначені для навчання, аналізу та

передбачення функціонування системи, повинні точно враховувати функціонування і поведінку існуючих систем військового призначення. Такі програмні продукти можуть використовуватися для обліку життєво важливих елементів поведінки військовослужбовця в бою, аналізу багатомільйонних рішень про закупівлю озброєння і військової техніки або визначення майбутньої організаційно-штатної структури збройних сил. Якщо вони не точні, результати можуть бути катастрофічними. Тому будь-які програмні продукти військового призначення повинні бути засновані на відомих особливостях поведінки реальних систем військового призначення. Зазвичай вихідних даних для опису функціонування таких систем недостатньо. Під час реальної війни акцент на збір даних об'єктивний для осіб, які приймають рішення на ведення бойових дій. Він обумовлений необхідністю виконання поставленого бойового завдання з найменшими втратами. В результаті у розробників програмних продуктів є дуже обмежений набір вимірної інформації про те, як протікає бій (операція).

Досвідчені розробники повинні знати все про бази даних, які існують у предметних областях, що описують системи військового призначення. Вони повинні зрозуміти, які дані є і доступні, а які дані повністю недоступні. Кожен програмний продукт вимагає даних, які не існують ні в якій офіційній чи неофіційній формі. Кожна модель вимагає, щоб дані були синтезовані з того, що доступно, а також на основі суб'єктивних думок військовослужбовців, які брали участь у бойових діях. При цьому зусилля при створенні програмного продукту, що має недостатньо вихідних даних, повинні бути зосереджені на пошуку відсутніх даних, які у реальності існують.

7. Зробіть можливим управління вихідними даними програмного продукту.

У минулому потужності центрального процесора і обмежена кількість оперативної пам'яті диктували вкрай низьку можливість використання складних програмних продуктів (моделей, військових ігор). В результаті всі вони були побудовані з алгоритмів, які описували певну ситуацію, для моделювання якої буде використовуватися даний програмний продукт. Зміна ситуації вимагало зміни програмного продукту.

Однак завдяки промислому розвитку апаратних засобів ми можемо тепер змінювати модель функціонування системи відповідно до зміни вихідних даних, які можуть бути змінені програмними командами або користувачем. Це дає можливість створювати програмні продукти, які набагато більш гнучкі й цим цінні користувачеві.

Навіть військові ігри тепер дозволяють користувачеві створювати свої власні сценарії, додавати нові блоки і змінювати візуальні сцени реального світу. Наявність таких можливостей програмного продукту - одне з актуальних питань користувача. Воно може бути вирішене лише у разі надання можливості користувачеві (не програмісту) безпечно керувати будь-якими доступними даними.

8. Довіряйте своєму творчому потенціалу.

Розробники, які створюють програмний продукт вперше, бояться їх розробляти без уявного розуміння того, що вони повинні створити. Вони бояться, що будуть рухатися в неправильному напрямку і створювати програмний продукт, який буде підданий критиці іншими. Цей страх перед критикою завдає більшої шкоди, ніж їх небажання до переробки програмного продукту, робота якого пішла не так як треба.

Досвідчені розробники повинні продемонструвати, прищепити і заохотити використання програмістами творчих здібностей. Керівники програмних проектів повинні забезпечити бачення всього програмного продукту в цілому, але кожен програміст повинен проявляти свободу творчості і впевненість, щоб висловити їх бачення в програмному продукті. При створенні проектів військового призначення страх перед допущенням помилок призводить до постійного повторення аналізу потреб до розробляемого продукту, організаційної реструктуризації, дослідженню продукту і неефективних зустрічей із замовником. Розробники уникають прийняття конкретних рішень щодо дизайну продукту. Це не дозволяє програмістам створити коректну концептуальну модель або збудувати прототип. Тисячі людино-годин можуть бути витрачені марно в цій пастці. Але в кінцевому рахунку цей цикл буде порушений однією з таких подій:

поява рішучого і мотивованого лідера;

загін розробки програмного продукту в глухий кут програмістами, які ігнорують вимоги керівника проекту;

жорсткі терміни закінчення розробки програмного продукту;

відміна проекту.

Досвідчені керівники проектів не дозволять опинитися в такий пастці. Досвідчені програмісти не будуть коливатися з вирішенням проблеми, яку вони знають як вирішити. Упевнені в собі програмісти (початківець і досвідчений) зможуть обійти цю пастку самостійно. Якщо розробники не довіряють своїм власним творчим здібностям, то вони або мають низьку кваліфікацію, або слабого керівника команди розробників.

9. Дотримуйтесь універсальних обмежень.

Кожен програмний продукт обмежений універсальними обмеженнями:

якість;

час;

вартість;

компетентність розробників.

Коли розробники домагаються виконання кожного з них, програмний продукт вважається закінченим.

Якість характеризує деталізацію і можливість моделювання програмного продукту. Час, необхідний для розробки програмного продукту визначає рівень його якості. Вартість програмного продукту буде насамперед визначатися кількістю розробників і пов'язана безпосередньо з фактором часу (так як всі будуть отримувати зароблені за

певний час гроші). Ці три обмеження застосовані до кожного програмного продукту в цілому.

Однак є четверте обмеження – компетентність розробників. Деякі проекти вимагають навичок програмістів, які знаходяться у дефіциті. Тому, чудово профінансований проект з великим часом створення може бути не реалізований у зв'язку з неможливістю залучити для виконання проекту програмістів з певними навичками, які здатні виконати цю роботу. Досвідчених лідерів, програмістів, проєктантів іноді не вистачає, щоб виконати всю роботу, яку визначив замовник.

Вдалих проект завжди повинен вписатися в межі, сформовані всіма чотирма з наведених вище обмежень.

10. Формуйте свої власні принципи.

Слід підкреслити, що існує багато більше, ніж десять принципів моделювання. Дев'ять, перерахованих вище принципів, були отримані на основі аналізу роботи досвідчених програмістів зі створення програмних продуктів різного призначення, в тому числі і військового. Однак кожен програміст має свій власний досвід створення програмних продуктів. Тому кожен програміст може виробити власний набір принципів, які він буде використовувати, щоб уникнути помилок, які були зроблені в минулому. Це дозволить реалізовувати в перспективі тільки вдалих проекти.

Правила збору необхідних даних для створення програмного продукту.

Будуючи віртуальну версію реального світу важливо мати можливість описати реальний світ в деякій числовій або заснованій на правилах формі, що дозволить надалі описати їх програмними кодами. Так як модель - це динамічна картина поведінки системи, то дуже важко оцінити наскільки точно вона описує цю поведінку. Дані про ініціалізацію, які запускають систему, є одним з індикаторів того, наскільки точно вона буде моделюватися. Це, звичайно, не єдиний і навіть не найсильніший індикатор. Але це легко вимірний і відчутний індикатор. Коли модель або гра запускані, події розвиваються занадто швидко. У цьому випадку важко забезпечити їх адекватність.

Військові моделі можуть бути набагато більш вибагливими до точності використовуваних даних, ніж комп'ютерні ігри. Але оскільки віртуальні світи в іграх стають більш реалістичними і більш інтерактивними, потреба в точних даних, які у розумних межах описують реальний світ, збільшується. Користувачі отримують більше відчуття реалізму при моделюванні, заснованому на взаємодії об'єктів моделювання, ніж при його статичному відображенні (у формі кількісних показників або скрін-шотів). Створення правдоподібних або реалістичних моделей можливо при обліку всіх особливостей поведінки і взаємодії об'єктів реального світу. Процес збору точних даних, за допомогою яких можна

побудувати модель реального світу, дозволяє визначити чотири правила їх збирання.

Перше правило. Ніколи не можна отримати всі необхідні дані про систему (об'єкт), яких ви потребуєте.

Незалежно від рівня програмного продукту, який створюється, повний комплект даних для його реалізації зібрати не вдається, а тим більше організувати і каталогізувати їх, щоб в повному обсязі задовольнити потреби розробника програмного продукту. Зібрані дані, які зазвичай доступні, отримують в ході дослідження властивостей моделі системи (об'єкта) або військової гри, яка розробляється. В результаті отримані дані ніколи не охоплюють всі аспекти створюваного віртуального світу.

Друге правило. Неможливо в повному обсязі використовувати всі дані, які можна отримати.

Перше правило не говорить про те, що дані по будь-якій системі (об'єкту) недостатні. Фактично світ наповнений інформацією про необхідні дані. Однак більша частина цієї інформації повторюється і суперечлива. Цей аспект проблеми також досить важливий, тому що часто позбавляє можливості об'єднувати дані з багатьох джерел в єдиний повний опис системи (об'єкта).

Третє правило. Завжди необхідно визначити деякі свої власні дані.

Перші два закони роблять третій закон цілком зрозумілим, так як розробники опиняються перед необхідністю робити збір деяких даних самостійно.

Четверте правило. Завжди необхідно синтезувати дані, щоб в повному обсязі задовольнити потреби у даних для створення програмного продукту.

Незважаючи на величезне бажання зібрати всі необхідні дані про поведінку системи (об'єкта) є деякі дані, які неможливо визначити або виміряти.

Кожен програмний продукт містить дані, які синтезуються досвідченими розробниками. Синтезування може бути засноване на законах фізики, екстраполяції експериментів, висловлювання припущень або бути чистою фантазією. Ніякий програмний продукт не може бути розроблений без творчої ініціативи розробників, готових висловлювати свої припущення та ідеї.

Висновки й перспективи подальших досліджень

У статті автори спробували узагальнити досвід створення програмних продуктів військового призначення за останні роки і узагальнити уроки, витягнуті з цього. Автори і ті, з якими їм довелось спілкуватися єдині у думці, що шлях до удачі або невдачі лежить через набутий досвід. Всі ми також вчилися на досвіді інших. Автори сподіваються, що розробники програмних продуктів отримають користь з викладеного у статті матеріалу і реалізують більше вдалих проєктів по створенню програмних продуктів військового призначення.

Література

1. Бирюков Б. В., Гастеев Ю. А., Геллер Е. С. Моделирование. - М.: БСЭ, 1974 г. 2. Бусленко Н. П.

Моделирование сложных систем. Москва, 1961. 3. Карпов Ю. Г. Имитационное моделирование систем.

Введение в моделирование с Anylogic 5. СПб.: БХВ-Петербург, 2009. 4. **Сорокин В. П.**, Пермяков А. Ю., Маслюк Л. А., Бобылев В. Е. и др. Моделирование боевых действий войск. Математические методы обоснования решений и оценки эффективности боевых действий войск. Ч.1. Учебное пособие. Коллектив авторов. К., АВСУ, 1995 г., 218 с. 5. **Сорокин В. П.**, Пермяков А. Ю., Маслюк Л. А., Бобылев В. Е. и др. Моделирование боевых действий войск. Методы моделирования и математические модели боевых действий войск. Ч.2. Учебное пособие. Коллектив

авторов. К., АВСУ, 1995 г., 256 с. 6. **Ткаченко П. Н.**, Куцев Л. Н., Мещеряков Г. А., Чавкин А. М., Чебыкин А. Д. Моделирование боевых действий войск. М., изд. "Советское радио", 1969, 240 стр. 7. **Шеннон Р.** Имитационное моделирование систем: искусство и наука. М., изд. "Мир", 1978, 418 с. 8. **Hughes**, Wayne P. Editor. 1997. Military Modeling for Decision Making, Third Edition. Military Operations Research Society, Alexandria, Virginia. 9. **Law**, Averill and Kelton, W. David. 1991. Simulation Modeling and Analysis. McGraw Hill. New York, NY.

ПРИНЦИПЫ МОДЕЛИРОВАНИЯ

Виктор Евгеньевич Бобылёв (канд. воен. наук, с.н.с.)

Юрий Николаевич Зинченко (канд. техн. наук)

Сергей Николаевич Кононенко

Национальный университет обороны Украины имени Ивана Черняховского, Киев, Украина

Создание интерактивных миров в компьютерных играх, интернет-пространствах и моделях военного назначения в последнее время стало хорошо налаженным процессом, основанным на опыте отдельных проектировщиков и программистов включительно. Однако с появлением все большего количества таких моделей и игр формируются принципы (основополагающие, основные и др.) их реализации, которые характерны для всех проектов подобного рода. В статье описываются основные принципы создания программных продуктов, которые получены путем анализа развития процессов создания программных продуктов предназначенных для имитации боевых действий войск и военных игр.

Ключевые слова: компьютерная игра; модель; программный продукт; принцип; правило; исходные данные; моделирование.

THE MODELING PRINCIPLES

Victor Y. Bobylov (Candidate of Military Sciences, Senior Research Fellow)

Yurii M. Zinchenko (Candidate of Technical Sciences)

Serhii M. Kononenko

National Defence University of Ukraine named after Ivan Cherniakhovsky, Kyiv, Ukraine

Principles of modeling presented in this article were obtained from the analysis of years of experience creating computer models of military structures in the relevant ministries of defense around the world. Military Computer models are very similar to computer games and in some cases is the future of computer games. Therefore, the principles will be described below may be present and promising applicability for developers of software and games.

Create interactive worlds in computergames, the Internet spaces and models for military use has recently become well-established process based on the experiences of individual designers and programmers, inclusive. However, with the emergence of a growing number of such models and games formed the principles (fundamental, basic, etc.) For their implementation, which are characteristic for all projects of this kind. The article describes the basic principles of software products, which are obtained by the analysis of the processes of creation of software products designed to simulate combat operations and war games.

Keywords: computer game; model; a software product; the principles of the modeling; rule; the source data; modeling.

References

1. **Birjukov B.V.**, Gasteev Ju.A., Geller E.S. (1974), Modeling. [Modelirovanie], Moscow, BSJe. 2. **Buslenko N.P.** (1961), Modeling of Complex Systems. [Modelirovanie slozhnyh sistem], Moscow. 3. **Karpov Ju.G.** (2009), Simulation systems. Introduction to modeling Anylogic 5. [Imitacionnoe modelirovanie sistem. Vvedenie v modelirovanie s Anylogic 5], SPb, BHV-Peterburg. 4. **Sorokin V.P.**, Permjakov A.Ju., Masljuk L.A., Bobilev V.E. (1995), Modeling of combat operations. Mathematical methods support decisions and evaluate the effectiveness of combat operations. Part 1. Tutorial. [Modelirovanie boevykh dejstvij vojsk. Matematicheskie metody obosnovaniya reshenij i ocenki jeffektivnosti boevykh dejstvij vojsk. Ch.1. Uchebnoe posobie], Kyiv, AVSU, 218 p. 5. **Sorokin V.P.**, Permjakov A.Ju., Masljuk L.A., Bobilev V.E. (1995), Modeling of combat operations. Simulation methods and

mathematical models of combat operations. Part2. Tutorial. [Modelirovanie boevykh dejstvij vojsk. Metody modelirovanija i matematicheskie modeli boevykh dejstvij vojsk. Ch.2. Uchebnoe posobie], Kyiv, AVSU, 256 p. 6. **Tkachenko P.N.**, Kucev L.N., Meshherjakov G.A., Chavkin A.M., Chebykin A.D. (1969), Modeling of combat operations. [Modelirovanie boevykh dejstvij vojsk], Moscow, izd. "Sovetskoe radio", 240 p. 7. **Shennon R.** (1978), Simulation systems: the art and science. [Imitacionnoe modelirovanie sistem: iskusstvo i nauka], Moscow, izd. "Mir", 418 p. 8. **Hughes**, Wayne P. (1997), Editor. Military Modeling for Decision Making, Third Edition. Military Operations Research Society, Alexandria, Virginia. 9. **Law**, Averill and Kelton, W. David. 1991. Simulation Modeling and Analysis. McGraw Hill. New York, NY.

Отримано: 03.11.2015 року