Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Метод і методика формального проектування комплексної системи захисту інформації в інформаційно-телекомунікаційних системах
(тема)

Виконала: Гвоздьов Р. Ю.
(прізвище, ініціали)
студент 2 курсу, групи БІКСм-19-1
Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)

Керівник проф. Олійников Р.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Безпеки інформаційних технологій
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 125 Кібербезпека
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна, або освітньо-наукова)

Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові *Гвоздьову Роману Юрійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Метод передачі повідомлень з використанням квантово-захищеного криптографічного алгоритму*
затверджена наказом по університету від "22" жовтня 2020 р. № 1412Ст
2. Термін подання студентом роботи (проекту) 16.12.2020
3. Вихідні дані до роботи (проекту) вимоги нормативних документів в сфері технічного захисту інформації щодо процесу проектування комплексної системи захисту інформації
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)
1. Аналіз проблем розробки комплексної системи захисту інформації
2. Аналіз методів формального опису інформаційно-телекомунікаційної системи
3. Аналіз методів формального моделювання політики безпеки інформації
4. Розробка методики формального проектування комплексної системи захисту інформації
5. Реалізація формальної моделі політики безпеки інформації
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Презентаційний матеріал у вигляді слайдів*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської атестаційної роботи	Термін виконання етапів роботи	Примітка
1	*Отримання завдання*	*10.09.20*	Виконано
2	*Аналіз літературних джерел за темою атестаційної роботи*	*10.09.20-29.09.20*	Виконано
3	*Аналіз методів формального опису інформаційно-телекомунікаційної системи*	*29.09.20-11.10.20*	Виконано
4	*Аналіз методів формального моделювання політики безпеки*	*11.10.20-15.11.20*	Виконано
5	*Реалізація формального опису політики безпеки інформації*	*15.11.20-01.12. 20*	Виконано
6	*Оформлення пояснювальної записки*	*02.12.20-08.12. 20*	Виконано
7	*Представлення роботи на здачу*	*08.12.20-16.12.20*	Виконано

Дата видачі завдання 11 ___вересня_____2020 р.

Студент _____
(підпис)

Керівник роботи (проекту) _____ __проф. Олійников Р.В._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до роботи містить 84 с., 5 ч., 41 рис., 4 табл., 1 дод., 30 джерел.

КСЗІ, ІТС, НД ТЗІ, UML, UMLSEC, ДІАГРАМА, КОМПОНЕНТ, КОМПЛЕКС ЗАХОДІВ ЗАХИСТУ, РІВЕНЬ ГАРАНТІЙ

Мета роботи – розробка методики формального проектування КСЗІ, що включає в себе:

– формалізоване моделювання політики безпеки;

– формалізований опис ІТС та процесів обробки інформації;

– алгоритм формування комплексу засобів захисту у ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації.

Об’єкт дослідження – процес формального проектування КСЗІ ІТС.

Предмет дослідження – теоретичні, методичні засади та практичні аспекти реалізації формального опису ІТС та процесів розробки інформації та формальної моделі політики безпеки інформації.

Методи дослідження – в роботі для вирішення зазначених завдань було використано системний аналіз та програмування формальними мовами опису.

У роботі розглянуто методи та процес формального проектування комплексної системи захисту інформації.

Основні результати – продемонстровано формальну модель політики безпеки інформації.

ABSTRACT

The explanatory note contains: 84 pages, 5 parts, 41 figures, 4 tables, 1 addition, 30 sources.

CIPS, ITS, ND TPI, UML, UMLSEC, DIAGRAM, COMPONENT, SET OF PROTECTION MEASURES, LEVEL OF GUARANTEES

The purpose of the work is to develop a methodology for formal design of CIPS, which includes:

– formalized modeling of security policy;

– formalized description of ITS and information processing processes;

– algorithm of formation of a complex of means of protection in ITS from the formal model of security policy and from the formalized description of ITS and information processing operations.

The object of this research is the process of formal design CIPS in ITS.

The subject of the research is theoretical, methodological principles and practical aspects of the implementation of the formal description of ITS and information development processes and the formal model of information security policy.

Research methods - in the work to solve these problems was used systematic analysis and programming in formal description languages.

The methods and process of formal design of a complex information protection system are considered in the work.

The main results - a formal model of information security policy is demonstrated.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

UML	–	Unified Modelling Language
АС	–	Автоматизована система
ІТС	–	Інформаційно-телекомунікаційна система
КЗЗ	–	Комплекс заходів захисту
КС	–	Комп’ютерна система
КСЗІ	–	Комплексна система захисту інформації
НД	–	Нормативний документ
НСД	–	Несанкціонований доступ
ОС	–	Операційна система
ПЕОМ	–	Персональна електронно-обчислювальна машина
ПЗ	–	Програмне забезпечення
ПРД	–	Правила розмежування доступу
СЗІ	–	Служба захисту інформації
ТЗІ	–	Технічний захист інформації
ФПБ	–	Функціональна послуга безпеки

ВСТУП

Зі стрімким збільшенням кількості послуг, що надають інформаційно-телекомунікаційні системи (далі – ІТС), зростає складність архітектури ІТС. Недоліки при проектуванні таких систем можуть критично вплинути на їх функціонування, зокрема на безпеку. Якщо в ІТС планується оброблення інформації, порядок захисту якої регламентується законами України або іншими нормативно-правовими актами (наприклад, інформація, яка становить державну таємницю або вимоги до захисту якої встановлено законодавством), обов'язковим є незалежне підтвердження (оцінювання) відповідності реалізованих засобів та заходів захисту встановленим вимогам та нормам.

В Україні як критерії оцінки використовуються критерії, встановлені в [2]. Згідно з цими вимогами, оцінюються реалізовані функції захисту (функціональні послуги безпеки, ФПБ) та рівень гарантій коректності їх реалізації (рівень гарантій).

Рівень гарантій коректності реалізації ФПБ містять вимоги до архітектури комплексу засобів захисту (КЗЗ), середовища розробки, послідовності розробки, середовища функціонування, експлуатаційної документації та випробувань КЗЗ. Зокрема [11], вводиться сім рівнів гарантій (Г-1, ..., Г-7), які є ієрархічними. Ієрархія рівнів гарантій відображає поступово зростаючу впевненість у тому, що реалізовані в об’єкті інформаційної діяльності ФПБ дозволяють протистояти певним загрозам, а також, що механізми, які їх реалізують, у свою чергу коректно реалізовані та можуть забезпечити очікуваний споживачем рівень захищеності інформації під час її оброблення в ІТС.

Наприклад, для заявленого рівня гарантій Г-4 і більше реалізованої КЗЗ об’єкта необхідно викладати опис проекту архітектури у формалізованому вигляді, тобто використовуючи формальну нотацію. На даний момент часу не існує чітко визначеної методики для формального проектування КСЗІ в ІТС.

Обраний напрям дослідження висвітлює сучасні проблеми в сфері технічного захисту інформації в Україні. Однією з проблем, яка потребує розв'язання є відсутність методик для формального проектування комплексної системи захисту інформації в ІТС. Також відсутні наукові дослідження в даній тематиці.

Мета роботи – розробка методики формального проектування КСЗІ, що включає в себе:

– формалізоване моделювання політики безпеки;

– формалізований опис ІТС та процесів обробки інформації;

– алгоритм формування комплексу засобів захисту у ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації.

Об'єкт дослідження – процес формального проектування КСЗІ ІТС.

Предмет дослідження – теоретичні, методичні засади та практичні аспекти реалізації формального опису ІТС та процесів розробки інформації та формальної моделі політики безпеки інформації.

Методи дослідження – в роботі для вирішення зазначених завдань було використано системний аналіз та програмування формальними мовами опису.

Результати досліджень, що викладені у магістерській роботі, було оприлюднено на Харківському Безпековому Форумі 2020 під патронатом Урядового офісу координації європейської та євроатлантичної інтеграції Секретаріату Кабінету Міністрів України та за сприяння Харківської обласної державної адміністрації.

В тезах доповідей восьмої міжнародної науково-технічної конференції «Проблеми інформатизації», у всеукраїнському міжвідомчому науково-технічному збірнику «Радіотехніка» та в матеріалах Першого міжнародного науково-практичного форуму «Global Cyber Security Forum» опубліковані результати магістерської роботи.

1 ПРОБЛЕМАТИКА ПРОЕКТУВАННЯ КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Постановка задачі досліджень

Нормативними документами (далі – НД) в сфері технічного захисту інформації (далі – ТЗІ) визначено сім ієрархічних критеріїв гарантій від Г-1 до Г-7 включно, які визначають ступінь впевненості в тому, що кожна з функціональних вимог безпеки здатна протистояти певним загрозам. НД ТЗІ 2.5-004.99 висуває вимоги до процесу проектування КСЗІ, де стиль формалізованої (частково формалізованої) специфікації є обов'язковим для отримання рівня гарантій Г-4 та вище.

В таблиці 1.1 показані вимоги [3] до процесу проектування ,які забезпечують гарантії того, що на кожній стадії розробки (проектування) існує точний опис КС і реалізація КС точно відповідає вихідним вимогам (політиці безпеки).

Таблиця 1.1 Вимоги до процесу проектування КСЗІ

<table>
<tr><th>Вимоги</th><th>Г-1</th><th>Г-2</th><th>Г-3</th><th>Г-4</th><th>Г-5</th><th>Г-6</th><th>Г-7</th></tr>
<tr><td colspan="8">Функціональні специфікації (політика безпеки)</td></tr>
<tr><td>На стадії розробки технічного завдання Розробник повинен розробити функціональні специфікації КС. Представлені функціональні специфікації повинні включати неформалізований опис політики безпеки, що реалізується КЗЗ. Політика безпеки повинна містити перелік і опис послуг безпеки, що надаються КЗЗ</td><td>+</td><td>=</td><td>=</td><td>=</td><td>=</td><td>=</td><td>=</td></tr>
<tr><td colspan="8">Функціональні специфікації (модель політики безпеки)</td></tr>
<tr><td>Відповідність політиці безпеки</td><td rowspan="2">-

-</td><td colspan="2">Показ</td><td colspan="4">Демонстрація</td></tr>
<tr><td>Функціональні специфікації повинні включати модель політики безпеки</td><td>+</td><td>=</td><td>=</td><td>=</td><td>=</td><td>=</td></tr>
</table>

Продовження таблиці 1.1

<table>
<tr><td>Стиль специфікації: неформалізована</td><td>-</td><td></td><td></td><td colspan="3"></td></tr>
<tr><td>частково формалізована</td><td>-</td><td></td><td></td><td colspan="3"></td></tr>
<tr><td>формалізована</td><td>-</td><td></td><td></td><td colspan="3"></td></tr>
<tr><td colspan="8">Проект архітектури</td></tr>
<tr><td>Відповідність моделі політики безпеки</td><td>-</td><td colspan="2">Показ</td><td>Демонстрація</td><td colspan="2">Доказ</td></tr>
<tr><td>На стадії розробки ескізного проекту Розробник повинен розробити проект архітектури КЗЗ. Представлений проект повинен містити перелік і опис компонентів КЗЗ і функцій, що реалізуються ними. Повинні бути описані будь-які використовувані зовнішні послуги безпеки. Зовнішні інтерфейси КЗЗ повинні бути описані в термінах винятків, повідомлень про помилки і кодів повернення</td><td>+</td><td>=</td><td>=</td><td>=</td><td>=</td><td>=</td><td>=</td></tr>
<tr><td>Стиль специфікації: неформалізована</td><td colspan="2"></td><td colspan="3"></td><td colspan="2"></td></tr>
<tr><td>частково формалізована</td><td colspan="2"></td><td colspan="3"></td><td colspan="2"></td></tr>
<tr><td>формалізована</td><td colspan="2"></td><td colspan="3"></td><td colspan="2"></td></tr>
<tr><td>Вимоги</td><td>Г-1</td><td>Г-2</td><td>Г-3</td><td>Г-4</td><td>Г-5</td><td>Г-6</td><td>Г-7</td></tr>
</table>

На даний момент, не існує методик для формального проектування КСЗІ в інформаційно-телекомунікаційних системах (далі – ІТС).

Актуальною задачею на сьогоднішній день є аналіз існуючих мов формального опису системи, які в перспективі можуть використовуватися для проектування КСЗІ в ІТС, формування алгоритму комплексу засобів захисту у ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації та створення наукового підґрунтя для подальших досліджень в цій сфері.

1.2 Поняття та задачі формального проектування

Під терміном формалізованої специфікації слід розуміти таке представлення, яке базується на чітко визначених математичних концепціях. В свою чергу, математичні концепції визначають синтаксис і семантику подання, що дозволяє унеможливити неоднозначність розуміння моделі.

Процес проектування (або послідовність розробки) включає в себе модель політики безпеки та проект архітектури комплексу засобів захисту (далі – КЗЗ).

Методика формального проектування повинна включати:

– формалізоване моделювання політики безпеки;

– формалізований опис ІТС та процесів обробки інформації;

– алгоритм формування комплексу засобів захисту у ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації.

Основна задача формального проектування полягає у виборі методу формалізованого моделювання політики безпеки, методу формалізованого опису ІТС та процесів обробки інформації та формування алгоритму формування комплексу засобів захисту у ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації.

1.3 Вимоги при розробці формальних описів КСЗІ

Можна виділити такі основні вимоги щодо складу та подання проекту архітектури комплексу засобів захисту:

1) Опис усіх базових апаратних, програмно-апаратних та/або програмних засобів, що реалізують комплекс заходів захисту (далі – КЗЗ) з визначенням функцій механізмів захисту, реалізованими цими засобами;

2) мають бути визначені взаємозв'язки між всіма функціональними компонентами (підсистемами). Ці взаємозв'язки мають бути представлені на рівні зовнішніх інтерфейсів підсистем, потоків даних, керування тощо;

2) представлення інтерфейсів функціональних компонентів (підсистем) КЗЗ має містити опис призначення, методів використання інтерфейсів, повідомлень про помилки та кодів повернення, забезпечуючи, де це необхідно, деталізацію результатів та можливих позаштатних ситуацій (винятків);

3) проект архітектури має містити опис порядку захищеного функціонування кожного компонента КЗЗ – опис будь-яких операцій функціонального компонента КЗЗ, дії якого можуть спричинити зміну захищеного стану об’єкту, у вигляді послідовності дій, які виконуються в кожній підсистемі КЗЗ, як результат впливу на відповідний інтерфейс;

4) КЗЗ має викладатися в термінах послідовностей дій, які виконуються в кожному функціональному компоненті (підсистемі) КЗЗ у відповідь на ініціюючий вплив на відповідний інтерфейс;

5) мають бути описані всі використовувані зовнішні послуги безпеки (послуги, реалізовані функціональними компонентами, що не входять до складу КЗЗ ОЕ).

Для заявлених рівнів гарантій Г-6 … Г-7 проект архітектури має бути викладений у формалізованому вигляді, тобто, опис порядку захищеного функціонування компонентів (підсистем) КЗЗ має бути викладений з використанням формалізованої нотації, заснованої на чітко визначених математичних поняттях та супроводжуваної допоміжними поясненнями, наданими в неформалізованому вигляді. Використовувані математичні поняття мають забезпечувати визначення синтаксису та семантики представлення об'єктів КЗЗ, їх критичних для безпеки властивостей та виконуваних над ними операцій. У формалізованому описі мають бути відображені як послідовність, так і результати виконання різних операцій у функціональних компонентах (підсистемах) КЗЗ та всі пов'язані з їх виконанням виняткові або помилкові умови. Обов'язково має бути наведений (явно або у вигляді посилання) використовуваний набір домовленостей (правил), що визначають використовувану нотацію [5].

Для подання проекту архітектури у формалізованому вигляді (для заявлених рівнів гарантій Г-6 або Г-7), опис порядку захищеного функціонування компонентів КЗЗ має бути викладений згідно з попередньо-визначеними математичними поняттями. Пояснення математичних понять та використана нотація мають бути описані в неформалізованому вигляді. Мають бути визначені критичні властивості безпеки та виконувані над ними операції.

Для перевірки відповідності проекту архітектури та моделлю політики безпеки необхідно формально довести відповідність між захищеним функціонуванням компонентами КЗЗ та правилами політик реалізованих функціональних послуг безпеки (далі – ФПБ).

Інформація, яка міститься в цьому документі, повинна дозволити дійти висновку про те, що при розробленні проекту архітектури КЗЗ об’єкта інформаційної діяльності розробником адекватно та несуперечливо реалізовані вимоги моделі політики безпеки.

Для підтвердження такої адекватності та несуперечності в описі результатів аналізу відповідності між моделлю політики безпеки КЗЗ об’єкта інформаційної діяльності та проектом архітектури мають бути наведені аргументи, які підтверджують, що функціональними компонентами (підсистемами) КЗЗ об’єкта інформаційної діяльності в процесі їх захищеного функціонування згідно з порядком, викладеним у проекті архітектури, забезпечується реалізація правил та характеристик політик реалізованих ФПБ, викладених у моделі політики безпеки відповідного ступеня формалізації.

1.4 Критерії відбору методів

1.4.1 Критерії методу формалізованого опису ІТС та процесів обробки інформації

Для проектування систем використовують різні мови, різні підходи, тому необхідно ввести критерії та показники для відбору найкращих кандидатів з

ухилом на опис процесів безпеки та виконання вимог НД ТЗІ. Пропонується наступний перелік:

1) Складність. Показник складності характеризує, в першу чергу, здатність до адекватно-го опису потрібного параметру чи операції. Необхідність введення показника зумовлена тим, що при оцінюванні коректності реалізації рівня гарантій експерт може неправильно зрозуміти формальну модель, її формалізований вигляд або ж математичні концепції.

2) Орієнтованість на опис процесів обробки інформації. Під процесами обробки інформації найкраще тлумачення можна надати з [2] – виконання однієї або кількох операцій, зокрема: збирання, введення, записування, перетворення, зчитування, зберігання, знищення, реєстрації, приймання, отримання, передавання, які здійснюються в системі за допомогою технічних і програмних засобів.

3) Орієнтованість на опис процесів безпеки. Опис процесів безпеки можна поділити на чотири типи:

– конфіденційності;
– цілісності;
– доступності;
– спостереженості.

Кожна з чотирьох властивостей забезпечує захист від певної множини загроз.

4) Наявність інструментальної підтримки. Наявність готових програмних пакетів значно спрощує та прискорює процес розробки методики формального проектування. З іншого боку, готові інструменти для роботи можуть бути застарілі на даний момент, не підтримуватися розробниками чи багато коштувати.

1.4.2 Критерії методів формалізованого моделювання політики безпеки

Критерії для методу формалізованого моделювання політики безпеки:

– складність реалізації моделі політики безпеки;
– наявність інструментальної підтримки.

1.5 Вибір методу формалізованого моделювання політики безпеки

Існують наступні методи формалізованого моделювання політики безпеки:

– UMLSec [1, 9];

– Ponder2 [10].

Було здійснено порівняння методів формалізованого моделювання політики безпеки:

1) Порівняння методів формалізованого моделювання політики безпеки виконано за двома основними критеріями – наявність програмного забезпечення методу розробки та складність реалізації: UMLsec та Ponder мають в своєму складі готове програмне забезпечення для розробки;

2) реалізація вимог політики безпеки, які будуть засновані на логіці методу Ponder2, не є інтуїтивно-зрозумілими і, як наслідок, їх не легко відобразити механізмами мови. В свою чергу, UMLsec використовує базові механізми з відомого методу UML, що робить UMLsec більш привабливим методом формалізованого моделювання політики безпеки.

Отже, за критерієм складності UMLsec був обраний, як метод для формалізованого моделювання політики безпеки, бо має більш зрозумілу та чітку нотацію.

1.6 Вибір методу формалізованого опису ІТС та процесів обробки інформації

Існує один промислово розповсюджений метод опису – UML [7]. Інформаційна система в моделі UML зображується за допомогою основних елементів – компонентів, інтерфейсів та залежностей між ними. Формалізований опис ІТС подається у вигляді діаграми компонентів UML.

Діаграми UML доволі прості для розуміння після ознайомлення з його синтаксисом. Також існує можливість додавати власні текстові та графічні стереотипи, що значно розширює можливості застосування UML.

Завдяки розповсюдженості методу існує багато програмних середовищ для розробки UML-діаграм.

1.7 Алгоритм формування КЗЗ в ІТС з формальної моделі політики безпеки та з формалізованого опису ІТС та процесів обробки інформації

Вхідні дані алгоритму:

– діаграма компонентів UML (формалізований опис ІТС та процесів обробки інформації);

– формальний опис політики безпеки.

Діаграмою компонентів UML визначено вузли та інтерфейси системи. Усі інтерфейси кожного вузла перевіряються та висуваються необхідні вимоги політики безпеки. Проект архітектури КЗЗ, в підсумку, містить усі інтерфейси, правила їх взаємодії та вимоги політики безпеки.

2 АНАЛІЗ МЕТОДУ UML ПРИ ПОБУДОВІ МОЕДЛІ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНОЇ СИСТЕМИ

2.1 Структура мови UML

Концепції моделювання UML згруповані в мовні одиниці. Мовна одиниця UML складається з набору тісно зв'язаних концепцій моделювання, які надають користувачам можливість представляти аспекти системи, що проектується, відповідно до певної парадигми або формалізму [7]. Наприклад, одиниця мови UML State Machines дозволяє розробникам визначати дискретну поведінку, керовану подіями, використовуючи варіант відомого формалізму діаграм стану, тоді як мовна одиниця Activity забезпечує моделювання поведінки на основі парадигми, схожої на робочий процес. З точки зору користувача, це розділення UML означає, що їм потрібно застосовувати лише ті частини мови, які вони вважають необхідними для своїх моделей. Якщо ці потреби з часом змінюються, за необхідності до репертуару користувача можуть бути додані інші мовні одиниці. Отже, користувач UML не повинен знати повну мову, щоб ефективно використовувати її. Крім того, більшість мовних одиниць розділено на кілька розділів, кожен додає більше можливостей моделювання. Цей детальний розклад UML робить мову простішою для вивчення та використання, але окремі сегменти в цій структурі не представляють окремих точок відповідності. Остання стратегія призведе до перевищення показників відповідності та призведе до описаних вище проблем сумісності. Тим не менше, угруповання, що надаються мовними одиницями, та їх збільшення дійсно служать для спрощення визначення відповідності UML, як пояснено нижче.

Ця частина визначає статичні структурні конструкції [7] (наприклад, класи, компоненти, вузли), що використовуються в різних структурних діаграмах, таких як діаграми класів, діаграми компонентів та схеми розгортання. Пакети UML, які підтримують структурне моделювання показано на рисунку 2.1.

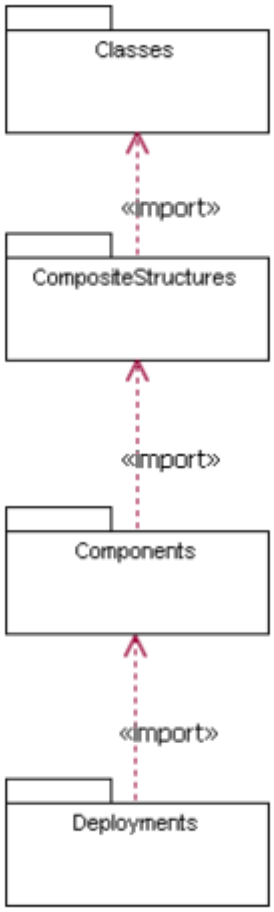

Рисунок 2.1 – Структура пакетів UML

Функції та вміст цих пакетів описані в наступних пунктах, які організовані за основними предметними областями.

2.1.1 Пакет ядра UML

Пакет "Класи" містить підпакети, що стосуються основних концепцій моделювання UML, зокрема класів та їх взаємозв’язки.

Пакет ядра представляє основні концепції моделювання UML [7], включаючи класи, асоціації та пакети. Ця частина здебільшого повторно використовується з бібліотеки інфраструктури, оскільки багато з цих понять є однаковими з тими, що використовуються, наприклад, у MOF. Пакет ядра є центральною частиною UML і повторно використовує пакети Constructs і PrimitiveTypes з InfrastructureLibrary. У багатьох випадках повторно використані класи розширюються в ядрі додатковими функціями, асоціаціями або

суперкласами. На наступних діаграмах [7], що демонструють абстрактний синтаксис, підклас елементів з бібліотеки інфраструктури завжди викреслюється, оскільки ця інформація лише додає складності без збільшення зрозумілості. Кожен метаклас повністю описаний як частина цього пункту; тут повторюється текст з бібліотеки інфраструктури. Слід також зазначити, що ядро – це плоска структура, яка, як і Constructs, містить лише метакласи, а не підпакети. Причиною такої різниці є те, що частини бібліотеки інфраструктури були розроблені для гнучкості та повторного використання, тоді як ядро при повторному використанні бібліотеки інфраструктури має об'єднати різні аспекти повторно використаних метакласів. Пакети, явно об’єднані з InfrastructureLibrary, є такими [7]:

– базові типи;

– конструкції;

Усі інші пакети InfrastructureLibrary::Core [7] неявно об’єднуються через ті, які явно об’єднані. На рисунку 2.2 зображено пакети бібліотеки InfrastructureLibrary, які об’єднані ядром (всі залежності на зображенні представляють злиття пакетів)

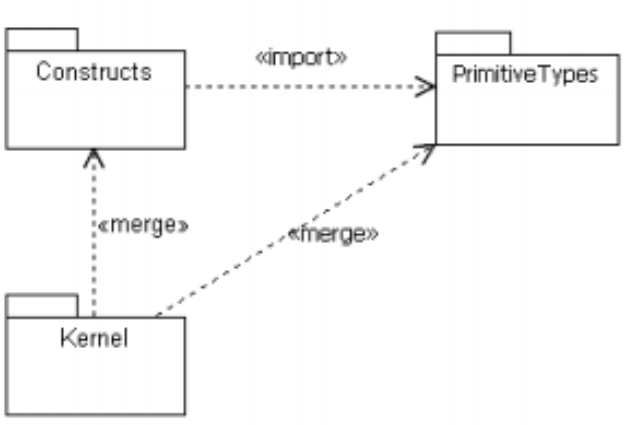

Рисунок 2.2 – Пакети бібліотеки InfrastructureLibrary

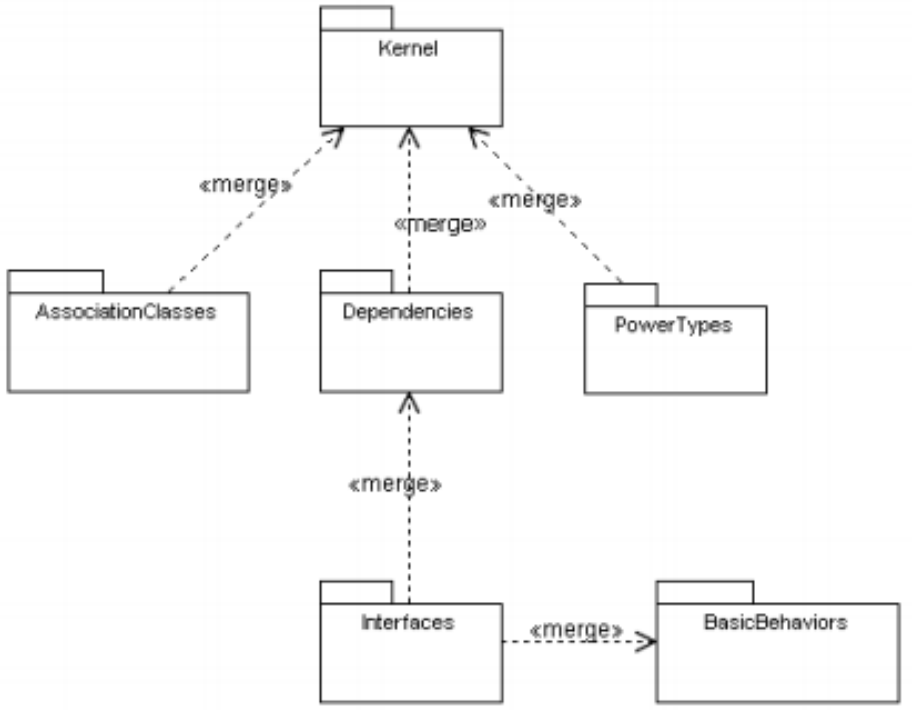

Рисунок 2.3 – Підпакети пакету "Класи" та їх залежності

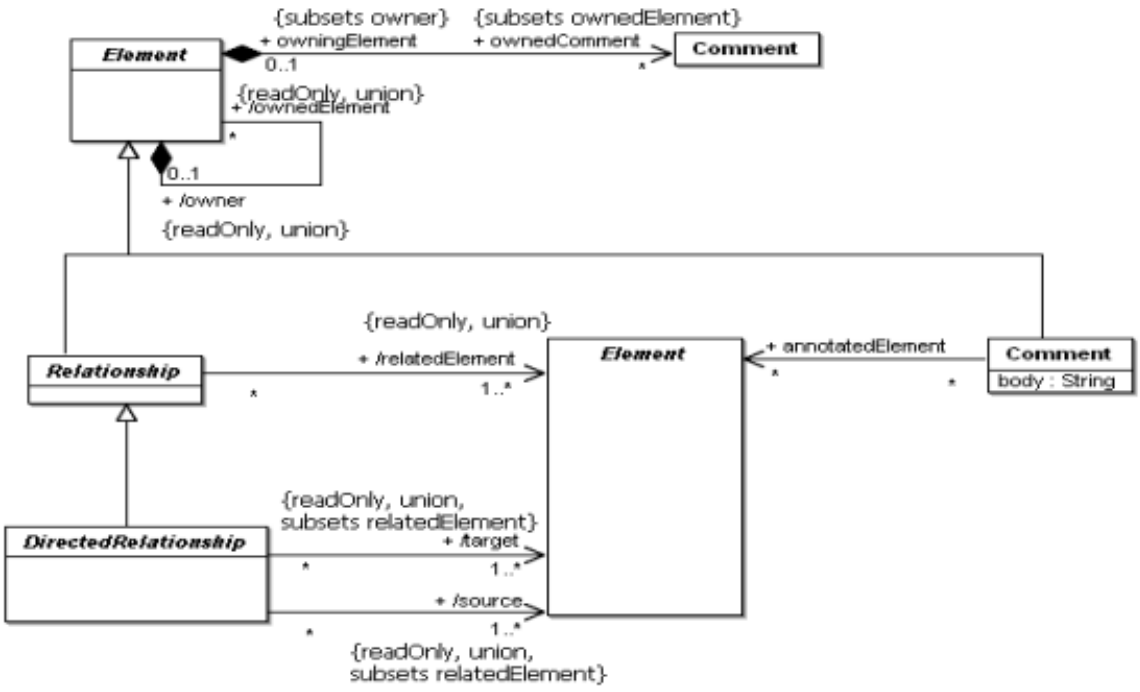

Рисунок 2.4 – Коренева діаграма пакета ядра

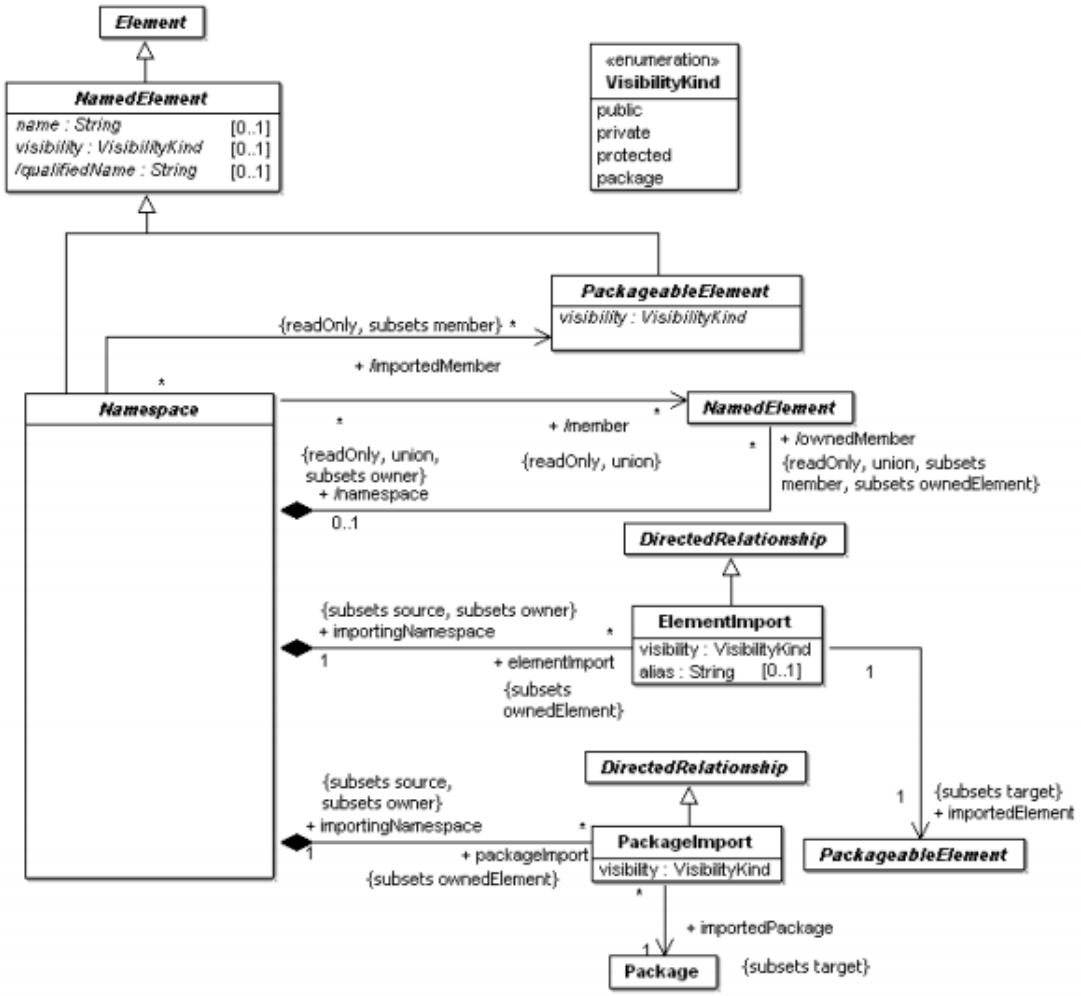

Рисунок 2.5 – Схема просторів імен пакета ядра

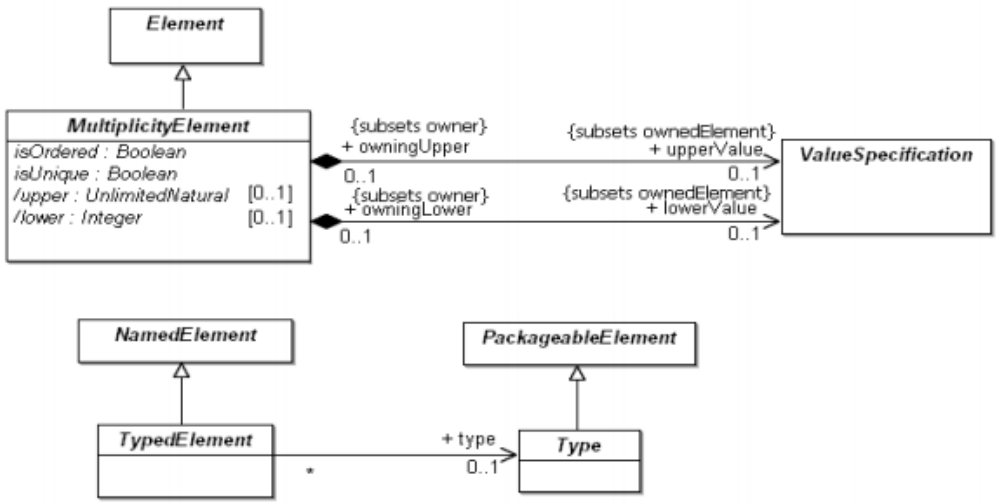

Рисунок 2.6 – Діаграма різноманітності пакета ядра

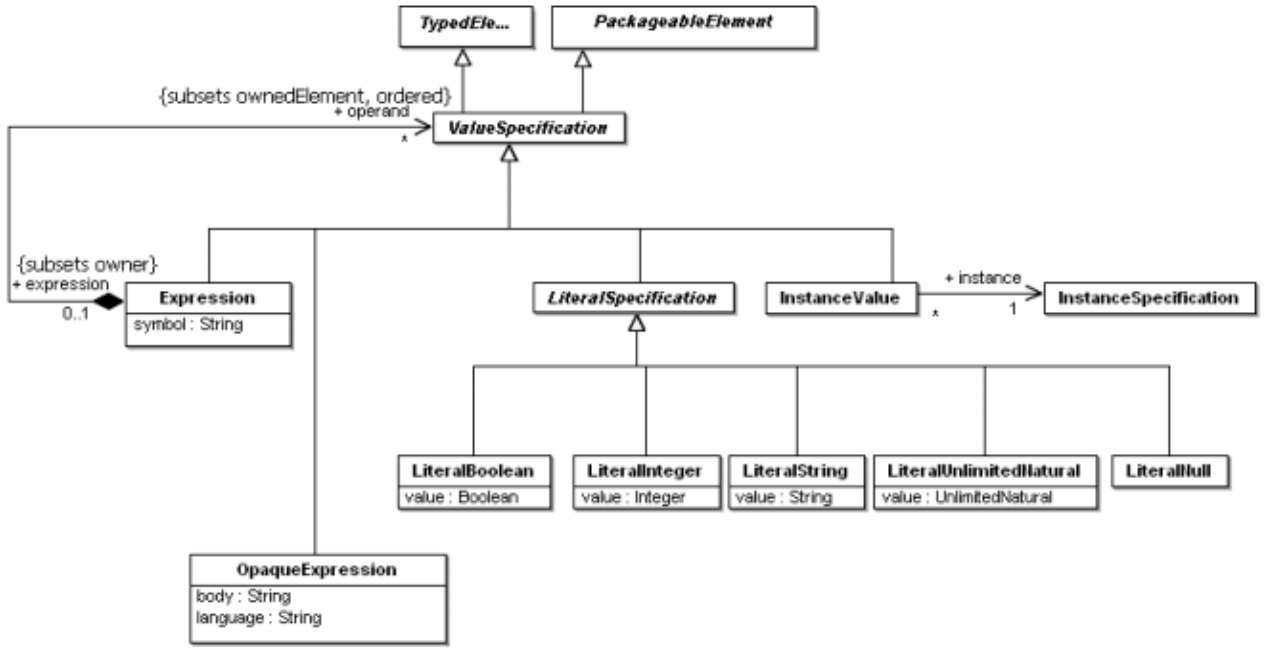

Рисунок 2.7 – Діаграма виразів пакету ядра

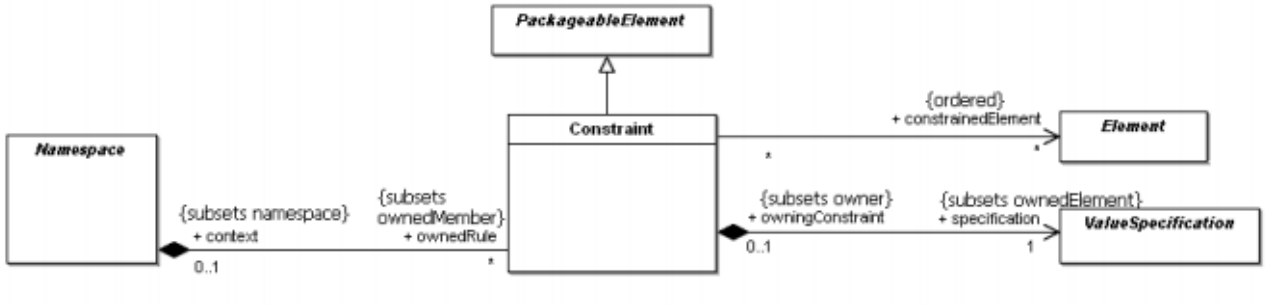

Рисунок 2.8 – Діаграма обмежень пакету ядра

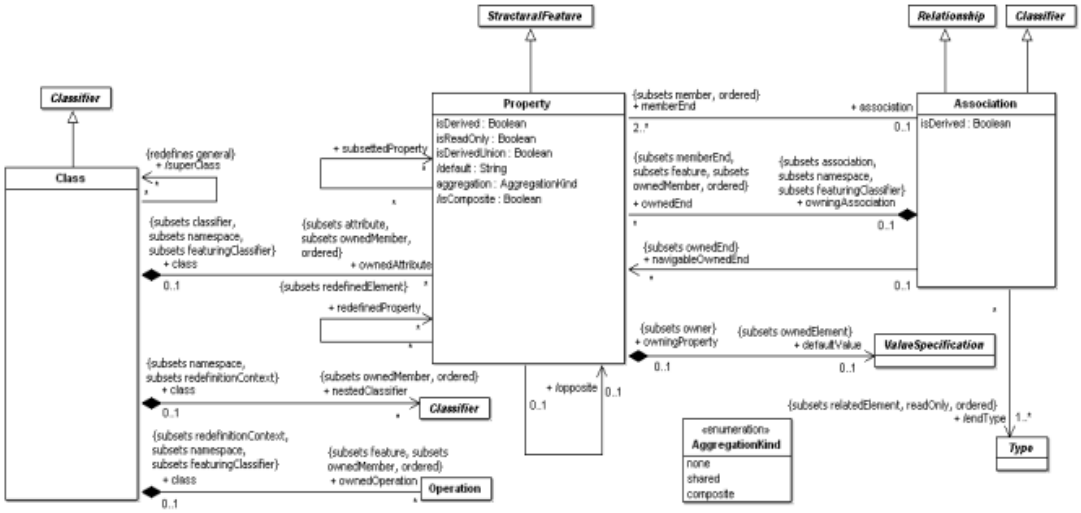

Рисунок 2.9 – Діаграма класів пакету ядра

2.1.2 Діаграми класів UML та їх нотація

Клас описує набір об’єктів, що мають однакові специфікації функцій, обмежень та семантики.

Клас – це різновид класифікатора, ознаками якого є атрибути та операції. Атрибути класу представлені екземплярами Property, які належать класу. Деякі з цих атрибутів можуть представляти навігаційні кінці двійкових асоціацій.

Асоціації [7]:

1) NestedClassifier: Класифікатор [*]. Посилається на всі Класифікатори, визначені (вкладені) в Клас. Підмножини Element :: ownMember

2) OwnerAttribute: Property [*]. Атрибути (тобто властивості), що належать класу. Асоціація замовлена. Класифікатор підмножин :: attribute та простір імен :: ownMember

4) OwnOperation: Operation [*]. Операції, що належать класу Асоціація. Класифікатор підмножин :: особливість і простір імен :: ownMember

5) SuperClass: Клас [*]. Це дає суперкласи класу. Він перевизначає Classifier :: general.

Призначення класу – вказати класифікацію об’єктів та визначити ознаки, що характеризують структуру та поведінку цих об’єктів. Об'єкти класу повинні

містити значення для кожного атрибута, який є членом цього класу, відповідно до характеристик атрибута, наприклад його типу та кратності. Коли екземпляр об'єкта створюється в класі, для кожного атрибута класу, який має вказане значення за замовчуванням, якщо початкове значення атрибута не вказано явно для екземпляра, тоді обчислюється специфікація значення за замовчуванням, щоб встановити початкове значення параметра атрибут для об’єкта. Операції класу можуть бути викликані на об'єкті з урахуванням певного набору підстановок для параметрів операції. Виклик операції може спричинити зміни значень атрибутів цього об'єкта. Він також може повернути значення як результат, де був визначений тип результату для операції. Виклики операцій також можуть спричинити зміни у значенні атрибутів інших об'єктів, до яких можна перейти, прямо чи опосередковано, від об'єкта, з яким викликається операція, до вихідних параметрів, до об'єктів, що переміщуються за її параметрами, або до інших об'єктів в обсязі виконання операції. Виклики операцій також можуть спричинити створення та видалення об'єктів. Клас не може отримати доступ до приватних функцій іншого класу або захищених об’єктів іншого класу, що не є його супертипом. Під час створення та видалення асоціацій принаймні один кінець повинен дозволяти доступ до класу [7].

Клас відображається за допомогою символу класифікатора. Оскільки клас є найбільш широко використовуваним класифікатором, ключове слово “клас” не потрібно показувати над назвою. Символ класифікатора без метакласу, позначає клас.

Клас часто показують із трьома відділеннями. Середній відсік містить список атрибутів, а нижній відсік – список операцій. Атрибути або операції можуть бути представлені згрупованими за видимістю. Потім ключове слово видимості або символ можна вказати один раз для кількох функцій з однаковою видимістю. Можуть бути надані додаткові відсіки, щоб показати інші деталі, такі як обмеження, або розділити елементи [7].

На рисунках 2.10 – 2.11 зображено приклади опису класів UML.

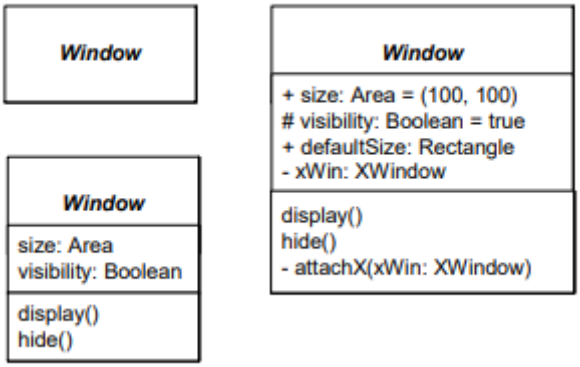

Рисунок 2.10 – Приклад опису класу UML

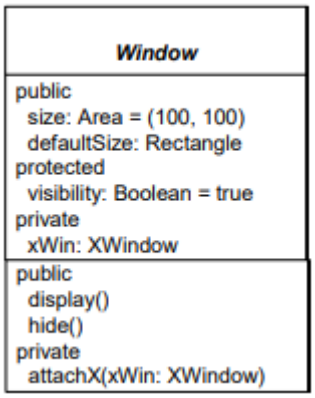

Рисунок 2.11 – Приклад опису класу UML

2.1.3 Діаграми розгортання UML та їх нотація

Пакет Deployments [6] визначає набір конструкцій, які можна використовувати для визначення архітектури виконання систем, що представляють призначення програмних артефактів вузлам. Вузли з'єднуються через комунікаційні шляхи для створення мережевих систем довільної складності. Вузли, як правило, визначаються вкладеним чином і представляють або апаратні пристрої, або середовища виконання програмного забезпечення. Артефакти представляють конкретні елементи у фізичному світі, які є результатом процесу розвитку. Пакет розгортання підтримує впорядковану модель розгортання, яка вважається достатньою для більшості сучасних програм.

Там, де потрібні більш складні моделі розгортання, їх можна розширити за допомогою профілів або метамоделей для моделювання конкретного апаратного та програмного середовища.

Схема розгортання показує архітектуру виконання систем, що представляють призначення (розгортання) програмних артефактів цілям розгортання (як правило, вузлів).

Вузли представляють або апаратні пристрої, або середовища виконання програмного забезпечення. Їх можна з'єднати за допомогою комунікаційних шляхів для створення мережевих систем довільної складності. Артефакти представляють конкретні елементи у фізичному світі, які є результатом процесу розробки та розміщуються на вузлах [6].

Варто звернути увагу, що компоненти були безпосередньо розгорнуті на вузлах у схемах розгортання UML 1.x. В UML 2.x [7] артефакти розгортаються до вузлів, і артефакти можуть проявляти (реалізовувати) компоненти. Тож компоненти тепер розгортаються на вузлах опосередковано через артефакти [6].

На діаграмі розгортання UML зазвичай малюються такі вузли та ребра: розгортання, артефакт, асоціація між артефактами, залежність між артефактами, компонент, прояв, вузол, пристрій, середовище виконання, склад вузлів, шлях зв'язку, специфікація розгортання, залежність від специфікації розгортання, асоціація специфікації розгортання.

Маніфестація [6] – це відношення абстракції, яке представляє конкретний фізичний візуалізацію (реалізацію) одного або декількох елементів моделі за допомогою артефакту або використання елементів моделі при побудові або генерації артефакту. Артефакт демонструє один або кілька елементів моделі.

Варто звернути увагу, що оскільки артефакти UML 2.0 можуть проявляти будь-які пакувальні елементи, а не лише компоненти, як це було в попередніх версіях UML [7].

Артефакт володіє проявами, кожен із яких представляє використання упакованого елемента.

Очікується, що конкретні профілі стереотипують взаємозв'язок проявів із зазначенням конкретних форм прояву. Наприклад, «згенерований інструмент» та «користувацький код» можуть бути двома проявами для різних класів, втілених у артефакті.

Маніфестація відзначається так само, як абстракція, тобто як пунктирна лінія з відкритою головкою стрілки, спрямована від артефакту до пакуваного елементу (наприклад, до компонента чи пакету), і позначається ключовим словом «маніфест».

Артефакт, визначений користувачем, являє собою конкретний елемент у фізичному світі. Конкретний екземпляр (або „копія“) артефакту розгортається на екземплярі вузла. Артефакти можуть мати асоціаційні композиції з іншими артефактами, вкладеними в нього. Наприклад, артефакт дескриптора розгортання для компонента може міститися в артефакті, який реалізує цей компонент. Таким чином, компонент та його дескриптор розгортаються на екземплярі вузла як один екземпляр артефакту. Очікується, що конкретні профілі створюють стереотипний артефакт для моделювання наборів файлів (наприклад, що характеризується „розширенням файлу“ у файловій системі). Стандартний профіль UML визначає кілька стандартних стереотипів, що застосовуються до Артефактів, наприклад, «джерело» або «виконуваний файл» (див. Додаток С - Стандартні стереотипи). Ці стереотипи можна додатково спеціалізувати на реалізації та специфічних для платформи стереотипах у профілях. Наприклад, профіль EJB може визначати «jar» як підклас «виконуваний файл» для виконуваних архівів Java [6].

Артефакт представлений за допомогою звичайного прямокутника класу з ключовим словом «артефакт». Як варіант, він може бути зображений піктограмою, як зображено на рисунку 2.12 та рисунку 2.13.

Необов’язково підкреслення імені екземпляра артефакту може бути опущено, оскільки контекст вважається відомим користувачів [7].

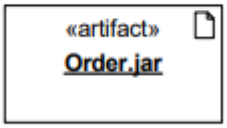

Рисунок 2.12 – Нотація артефакту

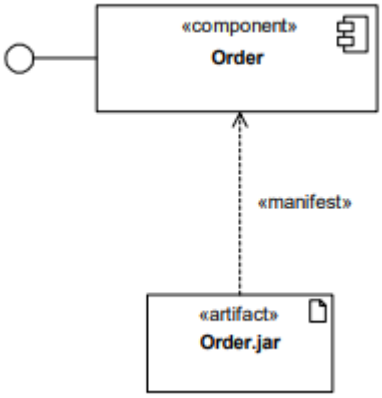

Рисунок 2.13 – Візуальне зображення взаємозв'язку проявів між артефактами та компонентами

Артефакти розгортаються для розгортання цілей. Ціль розгортання – місце розташування розгорнутого артефакту (рис. 2.14).

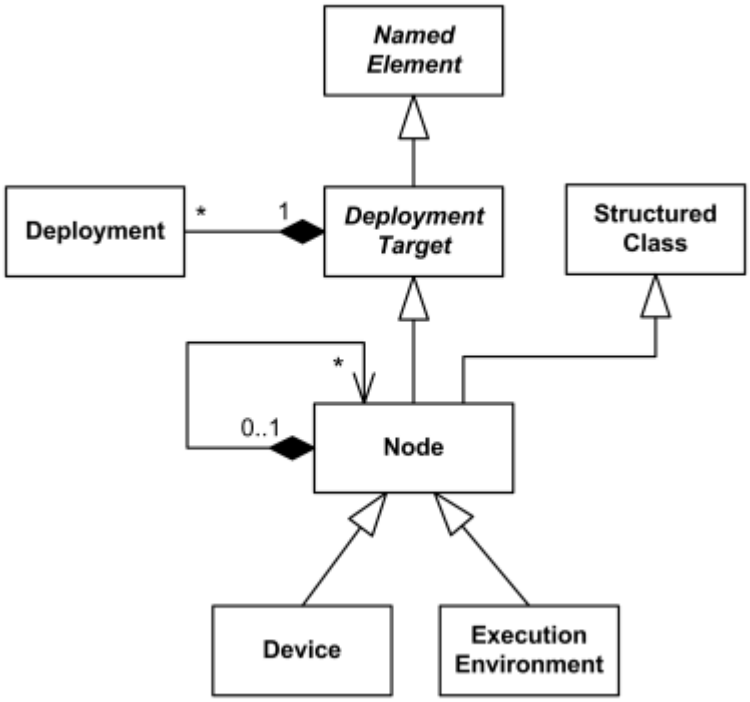

Рисунок 2.14 – Цілі розгортання UML 2.4

Специфікація екземпляра була розширена в UML 2.0 [7], щоб дозволити екземпляру вузла бути ціллю розгортання у відносинах розгортання.

Властивість також була розширена в UML 2.0 з можливістю бути ціллю розгортання у відносинах розгортання. Це дозволяє моделювати розгортання на ієрархічних вузлах, які мають властивості, що функціонують як внутрішні частини.

Вузол – ціль розгортання, яка представляє обчислювальний ресурс, на якому артефакти можуть бути розгорнуті для виконання.

Вузол відображається як тривимірний вигляд куба.

2.1.4 Діаграми компонентів UML

Компонент – це одиниця мови UML, яка інкапсулює стан і поведінку ряду класифікаторів. Компонент визначає офіційний контракт на послуги, які він надає своїм клієнтам, і ті, які він вимагає від інших компонентів або послуг у системі з точки зору своїх наданих та необхідних інтерфейсів. Компонент - це замінний блок, який може бути замінений під час проектування або виконання компонентом, який пропонує еквівалентну функціональність на основі сумісності своїх інтерфейсів. Поки середовище підпорядковується обмеженням, висловленим передбаченими та необхідними інтерфейсами компонента, воно зможе взаємодіяти з цим середовищем [8]. Подібним чином, систему можна розширити, додавши нові типи компонентів, які додають нову функціональність. Потрібні та надані інтерфейси компонента дозволяють вказати структурні особливості, такі як атрибути та кінці асоціації, а також особливості поведінки, такі як операції та події. Компонент може реалізовувати наданий інтерфейс безпосередньо, або його реалізаційні класифікатори можуть це робити, або вони можуть передаватися у спадок. Потрібні та надані інтерфейси за бажанням можуть бути організовані через порти, що дозволяють визначити іменовані набори наданих та необхідних інтерфейсів, на які зазвичай (але не завжди) звертаються під час виконання [7]. Компонент має зовнішній вигляд (або вигляд «чорної скриньки») завдяки своїм загальнодоступним властивостям та

операціям. За бажанням, поведінка, така як автомат стану протоколу, може бути приєднана до інтерфейсу, порту та самого компонента, щоб точніше визначити зовнішній вигляд, зробивши динамічні обмеження в послідовності викликів операцій явними. Інші способи поведінки також можуть бути пов'язані з інтерфейсами або роз'ємами для визначення "контракту" між учасниками співпраці (наприклад, з точки зору використання, діяльності чи специфікацій взаємодії). Структурне з'єднання між компонентами в системі або в іншому контексті може бути структурно визначено, використовуючи залежності між інтерфейсами компонентів (як правило, на структурних діаграмах) [7]. За бажанням, більш детальна специфікація структурної співпраці може бути зроблена з використанням деталей та з'єднувачів у складених структурах, щоб вказати роль або взаємодію на рівні екземпляру між компонентами (Див. Пункт Композиційні структури). Компонент також має внутрішній вигляд (або “білий ящик”) за допомогою своїх приватних властивостей та реалізації класифікаторів. Цей погляд показує, як зовнішня поведінка реалізується внутрішньо. Зіставлення між зовнішнім та внутрішнім видом здійснюється за допомогою залежностей (на структурних діаграмах) або з'єднання делегування з внутрішніми частинами (на складених структурних діаграмах). Знову ж таки, більш детальні специфікації поведінки, такі як взаємодії та дії, можуть бути використані для деталізації відображення зовнішньої поведінки на внутрішню. Існує ряд стандартних стереотипів UML, які застосовуються до компонента. Наприклад, «підсистема» для моделювання великомасштабних компонентів, а «специфікація» та «реалізація» для моделювання компонентів з чіткими визначеннями специфікації та реалізації, де одна специфікація може мати кілька реалізацій [7].

Компонент відображається у вигляді прямокутника класифікатора з ключовим словом «компонент». За бажанням, у правому куті може відображатися піктограма компонента. Це прямокутник класифікатора з двома меншими прямокутниками, що виступають з лівої сторони (рис. 2.15 та рис. 2.16).

Рисунок 2.15 – Нотація компоненту UML з одним наданим інтерфейсом

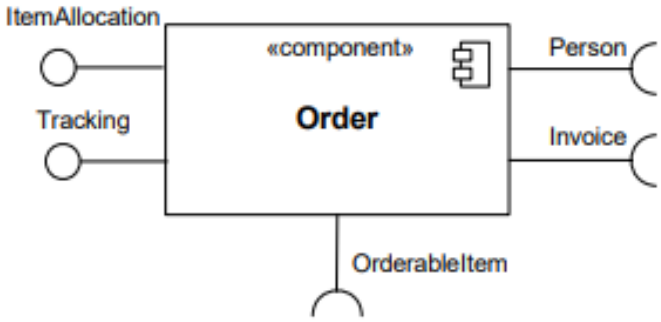

Рисунок 2.16 – Нотація компоненту UML із двома наданими та трьома необхідними інтерфейсами

Діаграма компонентів показує компоненти, надані та необхідні інтерфейси, порти та взаємозв'язки між ними. Цей тип діаграм використовується в розробці на основі компонентів (CBD) для опису систем із сервісно-орієнтованою архітектурою (SOA).

Розробка на основі компонентів базується на припущеннях, що раніше сконструйовані компоненти можуть бути використані повторно і що компоненти можуть бути замінені якимись іншими "еквівалентними" або "відповідними" компонентами, якщо це необхідно [8].

Артефакти, що реалізують компонент, мають змогу розгортати та перерозгортати самостійно, наприклад, для оновлення існуючої системи.

Компоненти в UML можуть представляти логічні компоненти (наприклад, бізнес-компоненти, компоненти процесів) та фізичні компоненти (наприклад, компоненти CORBA, компоненти EJB, компоненти COM + та .NET, компоненти WSDL тощо), разом із артефактами, що їх реалізують, та вузлами, на яких вони розгорнуті та виконані. Передбачається, що профілі на основі компонентів будуть

розроблені для конкретних компонентних технологій та відповідного апаратного та програмного середовища [8].

На діаграмі компонентів, як правило, малюються такі вузли та ребра: компонент, інтерфейс, наданий інтерфейс, необхідний інтерфейс, клас, порт, роз'єм, артефакт, реалізація компонента, залежність, використання. Ці основні елементи показані на рисунку 2.17.

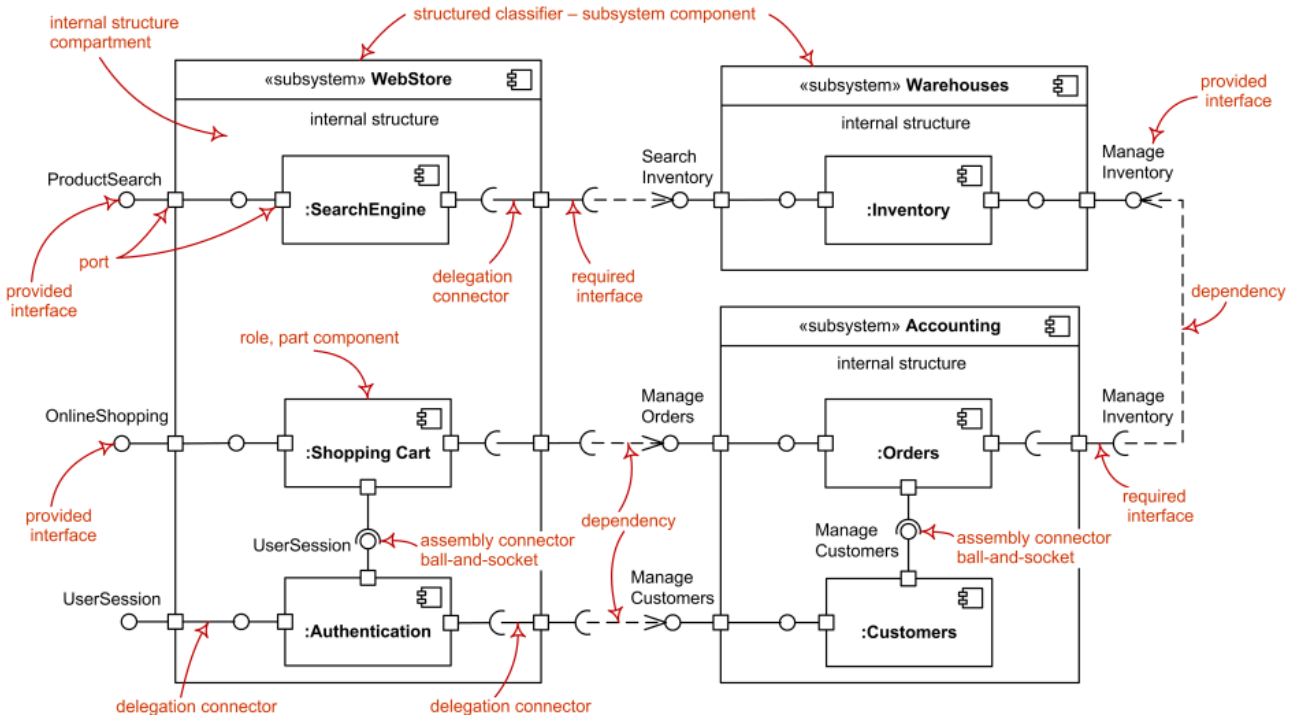

Рисунок 2.17 – Основні елементи діаграми компонентів UML

Для відображення повної сигнатури інтерфейсу компонента інтерфейси також можуть відображатися як типові прямокутники класифікатора, які можна розширити, щоб показати деталі операцій та подій (рис. 2.17).

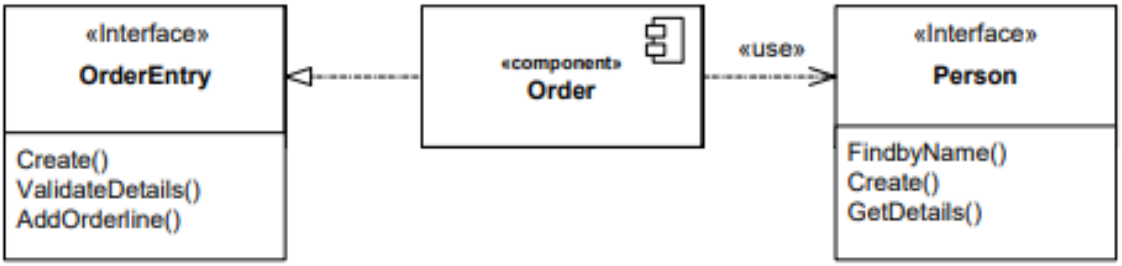

Рисунок 2.17 – Явне представлення наданих та необхідних інтерфейсів, що дозволяє відображати такі деталі інтерфейсу, як операція (опціонально)

Внутрішні класифікатори, що реалізують поведінку компонента, можуть відображатися за допомогою загальних залежностей [7]. Як варіант, вони можуть бути вкладеними у форму компонента (рис. 2.18).

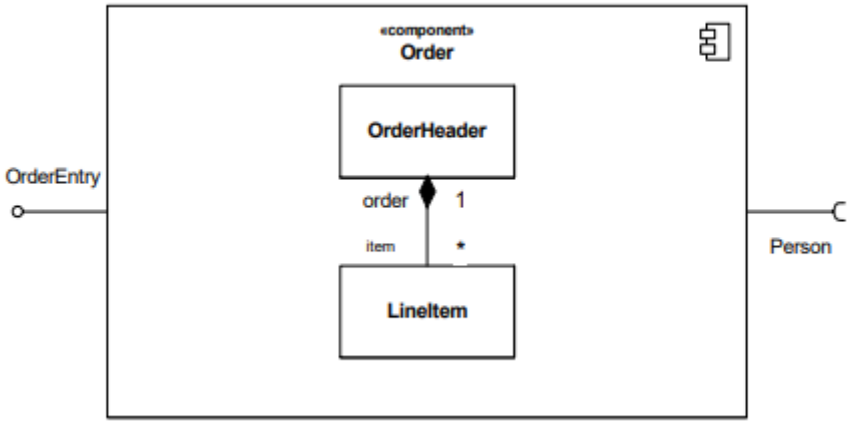

Рисунок 2.18 – Реалізація комплексного компоненту

Якщо потрібно більше деталей щодо вмісту компонента на рівні ролі чи екземпляра, тоді для цього компонента може бути визначена внутрішня структура, що складається з частин та з'єднувачів. Це дозволяє, наприклад, відображати явні назви деталей або назви сполучників у ситуаціях, коли один і той же класифікатор (асоціація) є типом більш ніж однієї деталі (з'єднувача). Тобто, класифікатор не раз створюється у компоненті, виконуючи різні ролі в його реалізації. За бажанням, на конкретні екземпляри (InstanceSpecifications) також можна посилатися як у цій нотації [7].

Інтерфейси, які виставляються компонентом і зазначаються на діаграмі безпосередньо або через визначення порту, можуть успадковуватися від компонента супертипу. Ці інтерфейси позначаються на схемі, перед назвою інтерфейсу косою рискою. Приклад цього можна знайти на рисунку 2.19, де “/ orderItem” – це інтерфейс, який реалізований супертипом компонента Product.

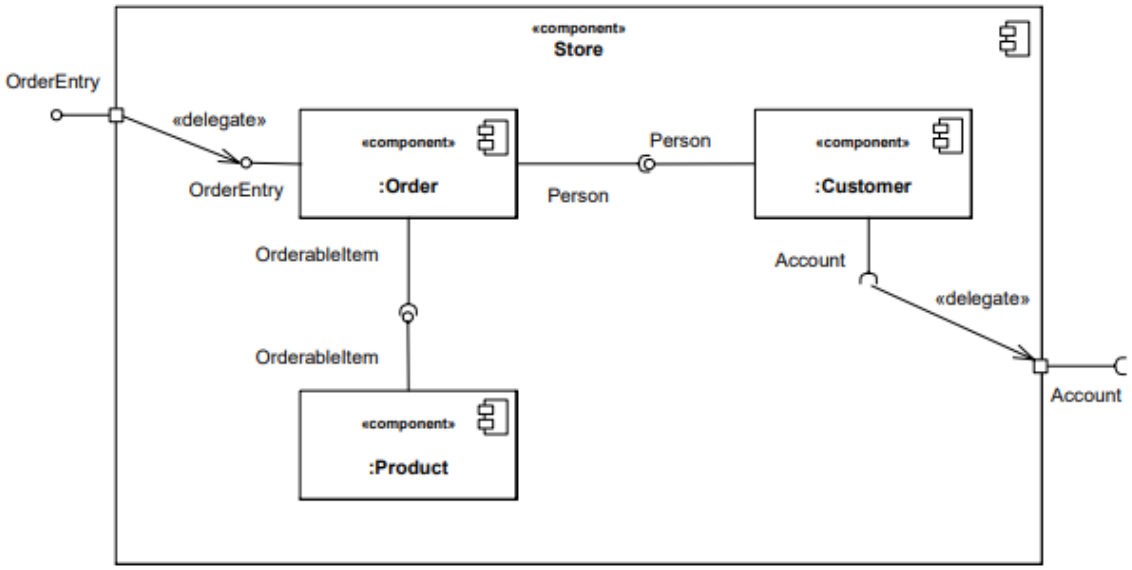

Рисунок 2.19 – Модель «білого ящику» внутрішньої структури компонента, який містить інші компоненти

Графічні вузли, які можуть бути включені в структурні схеми, наведені на рисунках 2.20 – 2.26.

Рисунок 2.20 – Варіанти нотації графічного вузла «компонент»

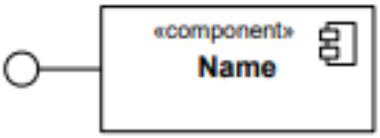

Рисунок 2.21 – Нотація графічного вузла «реалізація інтерфейсу компонента»

Рисунок 2.22 – Нотація графічного вузла «наданий інтерфейс компонента»

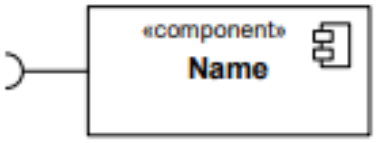

Рисунок 2.23 – Нотація графічного вузла «необхідний інтерфейс компонента»

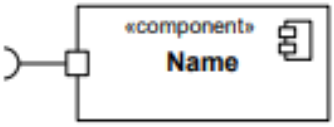

Рисунок 2.24 – Нотація графічного вузла «наданий порт компонента»

Рисунок 2.25 – Нотація графічного вузла «комплексний порт компонента»

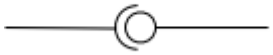

Рисунок 2.26 – Нотація графічного вузла «з'єднання інтерфейсів»

2.2 Вибір програмного середовища для розробки діаграм компонентів UML

Завдяки розповсюдженості методу існує велика кількість програмних середовищ для розробки моделей. В таблиці 2.1 наведено опис перспективних програмних пакетів для створення моделей UML.

Таблиця 2.1 – Порівняння інструментів розробки UML

Назва	Платформа, ОС	Відкритий код	Використана мова програмування
Eclipse UML2 Tools	Cross-platform (Java)	+	Java
JetUML	Cross-platform (Java)	+	Java
Papyrus	Windows, Linux, macOS (Java)	+	Java
PlantUML	Cross-platform (Java)	+	Java
UML Designer	Windows, macOS, Linux	+	Java, Sirius
Visual Paradigm for UML	Cross-platform (Java)	-	Java, C++
Modelio	Windows, Linux, macOS	+	Java

Продовження таблиці 2.1

Umbrello UML Modeller	Unix-like; Windows	+	C++, KDE

Всі визначені в таблиці 2.1 перспективні програмні інструменти станом на 2020 рік мають підтримку співтовариства розробників (у випадку відкритого коду) або ж компанії (у випадку платної ліцензії). В атестаційні роботі використовується програмне забезпечення Visual Paradigm for UML.

.

3 АНАЛІЗ МЕТОДУ UMLSEC ПРИ МОДЕЛЮВАННІ ПОЛІТИКИ БЕЗПЕКИ

3.1 Проектування з використанням нотації UMLsec

Основа ідея UMLsec – розширення існуючої моделі UML, шляхом додавання спеціальних міток – стереотипів, які додають відомості щодо безпеки. Відомості можуть бути наступного типу [1]:

– припущення щодо безпеки на фізичному рівні, наприклад як стереотип $||Internet||$;

– вимоги щодо безпеки на логічному рівні системи, наприклад як стереотип $||secrecy||$ (конфіденційність);

– вимоги політики безпеки, які накладаються на систему, наприклад як стереотипи $||secure\ links||$ (захищені зв'язки), $||no\ down\ flow||$ (керування потоком) .

Стереотип визначає новий тип елементів моделювання, розширюючи семантику вже існуючого типу або класу в моделі UML. Нотація стереотипу складається з імені стереотипу, взяті в подвійні прямі дужки || ||. Перелік стереотипів UMLsec наведено в таблиці 3.1.

Таблиця 3.1 – Перелік стереотипів UMLsec

Стереотип	Базовий клас	Тег	Обмеження	Опис
fair exchange	subsystem	start, stop, adversary	Після старту з часом досягне зупинки	Реалізація чесного обміну
provable	subsystem	action, cert, adversary	Незаперечна дія	Вимоги до відмов
rbac	subsystem	protected, role, right	Виконується тільки для дозволених дій	Реалізація контролю доступу на основі ролей

Продовження таблиці 3.1

Internet	link			Інтернет
encrypted	link			Зашифроване з’єднання
LAN	link, node			Локальна мережа
wire	link			кабель
smart card	node			Вузол смарт карти
POS device	node			POS-термінал
issuer node	node			Вузол постачальника
secrecy	dependency			Конфіденційність
integrity	dependency			Цілісність
high	dependency			Висока чутливість
critical	object, subsystem	Secrecy, integrity, authenticity, high, fresh		Критичний об’єкт
secure links	subsystem	adversary	Безпека залежностей відповідає посиланням «call», «send» відносно безпеки даних	Реалізація захищених ліній зв’язку
secure dependency	subsystem			Структурна взаємодія безпеки даних

Продовження таблиці 3.1

data security	subsystem	adversary, integrity, authenticity	Забезпечує конфіденційність, цілісність, автентичність, свіжість (новизна)	Базові вимоги до безпеки даних
no-down flow	subsystem	(data, origin)		Стан потоку інформації
no-up flow	subsystem	object name		Стан потоку інформації
guarded access	subsystem		Доступ до захищених об’єктів через механізми захисту	Контроль доступу з використанням захищених об’єктів
guarded	object	guard		Захищений об’єкт

Розширити модель [9] можна значенням тегів елементу моделі. Теги можуть розширити можливості при описі властивостей даних. Перелік тегів UMLsec наведено в таблиці 3.2.

Таблиця 3.2 – Перелік тегів UMLsec

Тег	Стереотип	Тип	Опис
start	fair exchange	state	Стан старту
stop	fair exchange	state	Стан зупинки
adversary	fair exchange	adversary model	Тип порушника

Продовження таблиці 3.2

action	provable	state	Операція/дія, що потребує підтвердження
cert	provable	expression	Сертифікат
adversary	provable	adversary model	Тип порушника
protected	rbac	state	Захищені ресурси
role	rbac	(actor, role)	Призначення ролі
right	rbac	(role, right)	Призначення прав до ролі
secrecy	critical	data	Конфіденційність даних
integrity	critical	(variable, expression)	Цілісність даних
authenticity	critical	(data, origin)	Автентичність даних
high	critical	message	Повідомлення високого рівня
fresh	critical	data	Нові дані
adversary	secure links	adversary model	Тип порушника
adversary	data security	adversary model	Тип порушника
integrity	data security	(variable, expression)	Цілісність даних
authenticity	data security	(data, origin)	Автентичність даних
guard	guarded	object name	Захищений об’єкт

Таким чином, використання нотації UMLsec дозволяє доповнити вже існуючу модель UML за допомогою надбудов безпеки. До реалізованих в UML нотацій, додаються параметри безпеки, які дозволяють реалізувати вимоги

політики безпеки та встановити відповідність між проектом архітектури та моделлю політики безпеки.

4 ОПИС ОБ’ЄКТА МОДЕЛЮВАННЯ

4.1 Вибір об’єкта моделювання

В якості демонстраційного прикладу формального моделювання політики безпеки було обрано інформаційно-телекомунікаційну систему веб-додатку (далі по тексту – веб-сторінка або веб-додаток), що реалізує доступ зовнішнім користувачам до інформаційної сторінки веб-серверу.

4.2 Призначення

ІТС веб-додатку призначена для забезпечення функціонування загальнодоступної веб-сторінки, яка адмініструється персоналом ІТС.

4.2 Склад автоматизованої системи

До складу АС, яка забезпечує функціонування веб-сторінки, входять: ОС, фізичне середовище, в якому вона знаходиться і функціонує, середовище користувачів, оброблювана інформація, у тому числі й технологія її оброблення. Під час забезпечення захисту інформації мають бути враховані всі характеристики зазначених складових частин, які впливають на реалізацію політики безпеки веб-сторінки. У випадку, якщо веб-сторінка містить посилання на інформаційні ресурси іншої веб-сторінки, умови функціонування останньої не повинні порушувати встановлену для даної веб-сторінки політику безпеки. У розділі 5 визначаються типові умови функціонування всіх складових АС, вводяться обмеження до умов функціонування та встановлюються загальні вимоги із захисту інформації до окремих компонентів АС. Для визначеної таким чином типової схеми функціонування АС встановлюються можливі варіанти для вибору функціональних профілів захищеності інформації від НСД.

До складу ІТС веб-додатку входить сервер.

4.3 Програмне забезпечення

До системного програмного забезпечення ІТС веб-додатку відносяться:

– ОС серверу з лінійки Linux;

– ПЗ веб-серверу Apache версії 2.2.

4.4 Середовище користувачів

Користувачі ІТС веб-додатку поділяються на наступні групи:

– внутрішні користувачі;

– зовнішні користувачі.

Внутрішні користувачі ІТС складаються з наступних категорій:

– адміністратор безпеки;

– адміністратор веб-сторінки.

Основними функціями адміністратора безпеки є:

– організація забезпечення виконання заходів з захисту інформації в ІТС веб-додатку;

– здійснення контрольних перевірок стану захищеності інформації в ІТС веб-додатку;

– керування параметрами безпеки КЗЗ програмного забезпечення;

– керування атрибутами доступу адміністраторів веб-сторінки;

аналіз журналів аудиту подій

– організація та контроль діяльності адміністраторів веб-сторінки.

Основними функціями адміністратора веб-сторінки є:

– редагування контенту веб-сторінки;

– координація діяльності із захисту інформації з адміністратором безпеки.

4.5 Можливі загрози інформації

Порушення конфіденційності, цілісності та доступності інформації, що обробляється у ІТС веб-додатку можуть проявлятися внаслідок таких загроз:

– порушення властивостей цілісності або доступності загальнодоступної інформації;

– порушення властивостей конфіденційності або цілісності технологічної інформації

– втрата атрибутів доступу користувачів ІТС веб-додатку, що призводить до неможливості використання функцій ІТС веб-додатку;

– несанкціоноване отримання або викривлення даних початкової ідентифікації та автентифікації користувачів ІТС веб-додатку;

– несанкціоноване використання обчислювальних ресурсів ІТС веб-додатку для досягнення цілей, які не відповідають призначенню ІТС веб-додатку;

– неправильне функціонування складових ІТС веб-додатку у наслідок порушення цілісності програмних засобів чи інших відмов;

– модифікація або видалення даних аудиту.

5 МЕТОДИКА ФОРМАЛЬНОГО ПРОЕКТУВАННЯ КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

5.1 Склад та вимоги до профілів захищеності інформації

Політика безпеки інформації в АС повинна поширюватися на об’єкти комп’ютерної системи, які безпосередньо чи опосередковано впливають на безпеку інформації.

До таких об’єктів належать:

– адміністратор безпеки та співробітники СЗІ;

– користувачі, яким надано повноваження забезпечувати управління АС;

– користувачі, яким надано право доступу до загальнодоступної

– інформації;

– інформаційні об’єкти, що містять загальнодоступну інформацію;

– системне та функціональне ПЗ, яке використовується в АС для

– оброблення інформації або для забезпечення функцій КЗЗ;

– технологічна інформація КСЗІ (дані про мережеві адреси, імена, персональні ідентифікатори та паролі користувачів, їхні повноваження та права доступу до об’єктів, встановлені робочі параметри окремих механізмів або засобів захисту, інша інформація баз даних захисту, інформація журналів реєстрації дій користувачів тощо);

– засоби адміністрування і управління обчислювальною системою АС та технологічна інформація, яка при цьому використовується;

– обчислювальні ресурси АС (наприклад, дисковий простір, тривалість сеансу роботи користувача із засобами АС, час використання центрального процесора і т. ін.), безконтрольне використання або захоплення яких окремим

користувачем може призвести до блокування роботи інших користувачів, компонентів АС або АС в цілому.

5.2 Функціональний профіль захищеності веб-додатку

В підрозділі 5.3 надається опис політики безпеки інформації з [4]. З урахуванням особливостей надання доступу до інформації веб сторінки, типових характеристик середовищ функціонування та особливостей технологічних процесів оброблення інформації, визначаються наступні мінімально необхідні рівні послуг безпеки для забезпечення захисту інформації від загроз:

– за умови, коли веб-сервер і робочі станції розміщуються на території установи-власника веб-сторінки або на території оператора (технологія Т1), мінімально необхідний функціональний профіль визначається {КА-2, ЦА-1, ЦО-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1, НЦ-1, НТ-1};

– за умови, коли веб-сервер розміщується у оператора, а робочі станції – на території власника веб-сторінки, взаємодія яких з веб-сервером здійснюється з використанням мереж передачі даних (технологія Т2), мінімально необхідний функціональний профіль визначається {КА-2, КВ-1, ЦА-1, ЦО-1, ЦВ-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1,НЦ-1, НТ-1, НВ-1}.

Технологія Т1 різниться від технології Т2 способом передачі інформації від робочої станції до веб-сервера, а саме: наявністю у другому випадку незахищеного середовища, яке не контролюється, і додатковими вимогами щодо ідентифікації та автентифікації між КЗЗ робочої станції й КЗЗ веб-сервера під час спроби розпочати обмін інформацією та забезпечення цілісності інформації при обміні.

За власником веб-сторінки залишається право реалізації, у разі необхідності, окремих послуг безпеки інформації зазначених профілів з більш високим рівнем, доповнення цих профілів іншими послугами, а також реалізація послуг безпеки з більш високим рівнем гарантій.

У випадках, коли в АС вимоги до політики реалізації якоїсь з послуг безпеки забезпечуються організаційними або іншими заходами захисту, які в повному обсязі відповідають встановленим НД ТЗІ 2.5-004 специфікаціям для певного рівня послуги безпеки, то рівень такої послуги, що входить до визначених згідно з 5.2 профілів захищеності, може бути знижений на відповідну величину.

5.3 Вимоги до реалізації функціональних послуг безпеки інформації

5.3.1 Базова адміністративна конфіденційність

КЗЗ повинен реалізувати рівень КА-2. Ця послуга дозволяє адміністратору безпеки керувати потоками інформації від захищених об'єктів до користувачів.

Політика адміністративної конфіденційності стосується: користувачів усіх категорій, об'єктів, що містять:

– технологічну інформацію КСЗІ та технологічну інформацію щодо управління АС;

– системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту загальнодоступної інформації та супроводження веб-сторінки;

– доступу користувачів до окремих видів периферійних пристроїв (принтерів, накопичувачів інформації тощо), використання яких передбачено технологією обробки інформації.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувача і захищеного об’єкта. Доступ до загальнодоступної інформації встановлюється для користувачів усіх категорій. Призначення атрибутів доступу користувачам і процесам до захищених об'єктів здійснюється адміністратором безпеки на основі аналізу функціональних та службових обов'язків окремих користувачів.

КЗЗ повинен надавати тільки адміністратору безпеки права доступу до технологічної інформації КСЗІ та процесів, що забезпечують її актуалізацію,

супроводження та аналіз. Доступ до процесів, що забезпечують ведення системних процесів з адміністрування й забезпечення функціонування АС в цілому, окремих її компонентів та сервісів, а також до технологічної інформації щодо управління АС повинен надаватись тільки користувачам, які мають відповідні повноваження.

Запити на зміну прав доступу повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністратора безпеки.

Права доступу до кожного захищеного об'єкта, визначеного політикою безпеки послуги, повинні встановлюватися в момент його створення або ініціалізації.

5.3.2 Конфіденційність при обміні

КЗЗ повинен реалізувати рівень КВ-1. Ця послуга дозволяє забезпечити захист об’єктів від несанкціонованого ознайомлення з інформацією, що міститься в них, під час їх експорту/імпорту через незахищене середовище.

Політика мінімальної конфіденційності при обміні стосується:

– користувачів, яким надано право супроводження КСЗІ та управління АС;

– об’єктів, які містять технологічну інформацію КСЗІ та технологічну інформацію щодо управління АС під час її передавання між віддаленими компонентами АС.

КЗЗ повинен забезпечувати захист від безпосереднього ознайомлення з інформацією, що міститься в об’єкті, який передається.

КЗЗ повинен забезпечувати можливість реєстрації подій, які призвели або можуть призвести до порушення конфіденційності інформації, що міститься в об’єктах, які передаються.

5.3.3 Мінімальна адміністративна цілісність

КЗЗ повинен реалізувати рівень ЦА-1. Ця послуга дозволяє керувати потоками інформації від користувачів до захищених об’єктів веб-сторінки.

Політика мінімальної адміністративної цілісності стосується:

– користувачів усіх категорій;

– загальнодоступної інформації веб-сторінки;

– файлової системи та функціонального ПЗ, що використовується для актуалізації, захисту загальнодоступної інформації та супроводження веб-сторінки;

– створеної в процесі супроводження веб-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС.

КЗЗ повинен здійснювати розмежування доступу на підставі атрибутів доступу користувачів і захищених об’єктів. Розмежування доступу здійснюється на рівні надання (встановлення заборони) користувачеві прав модифікувати об’єкт. Право визначати множину об'єктів АС, цілісність яких забезпечується КЗЗ, надається адміністратору безпеки.

КЗЗ повинен надавати можливість адміністратору безпеки для кожного захищеного об’єкта визначити домен, якому повинні належати ті користувачі і/або групи користувачів, що мають право модифікувати об’єкт. Тільки йому надається право включати і вилучати користувачів та об'єкти до/з конкретних доменів.

Призначення атрибутів доступу користувачам і процесам до захищених об'єктів та запити на зміну цих прав повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністратора безпеки. Користувачам, які мають доступ тільки до загальнодоступної інформації веб-сторінки, забороняється модифікувати будь-які захищені об’єкти. Адміністратору безпеки надається право модифікувати функціональне ПЗ, що використовується для захисту загальнодоступної інформації, та технологічну інформацію КСЗІ. Користувачам, що мають повноваження щодо управління АС, надається відповідно до функціональних обов’язків право модифікувати технологічну інформацію та функціональне ПЗ, що використовується для актуалізації загальнодоступної інформації та супроводження веб-сторінки.

Права доступу до захищених об’єктів веб-сторінки повинні встановлюватися в момент їх створення або ініціалізації.

5.3.4 Цілісність при обміні

КЗЗ повинен реалізувати рівень ЦВ-1. Ця послуга дозволяє забезпечити захист веб-сторінки від несанкціонованої модифікації інформації, яка передається між веб-сервером та робочими станціями у разі використання технології Т2, під час експорту/імпорту інформації через незахищене середовище. Політика послуги стосується всіх об’єктів, що передаються.

КЗЗ повинен забезпечувати контроль за цілісністю інформації в повідомленнях, які передаються, а також бути здатним виявляти факти їх несанкціонованого видалення або дублювання.

КЗЗ повинен забезпечувати можливість реєстрації подій, які призвели до порушення цілісності повідомлень, їх несанкціонованого видалення або дублювання.

5.3.5 Відкат

КЗЗ повинен реалізувати рівень ЦО-1. Ця послуга забезпечує можливість відмінити окрему операцію або послідовність операцій і повернути захищений об'єкт після внесення до нього змін до попереднього наперед визначеного стану.

Політика обмеженого відкату стосується користувачів, яким надано право супроводження КСЗІ та управління АС; об’єктів, які містять публічну інформацію; функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження веб-сторінки; створеної в процесі супроводження веб-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС. Якщо стосовно якогось з об’єктів зазначених категорій в процесі обробки не передбачається можливості його модифікації, політика послуги на нього не розповсюджується.

До складу АС повинні входити автоматизовані засоби, які дозволяють адміністратору безпеки, співробітнику СЗІ, користувачу, який має повноваження щодо управління АС, відкатити або відмінити певний набір (множину) операцій, виконаних над захищеним об’єктом веб-сторінки за певний проміжок часу.

Факт використання послуги має реєструватись в системному журналі. Відміна операції не повинна призводити до видалення з журналу запису про

операцію, яка пізніше була відмінена, якщо остання підлягала реєстрації відповідно до вимог послуги безпеки НР-2.

5.3.6 Використання ресурсів

КЗЗ повинен реалізувати рівень ДР-1. Ця послуга дозволяє керувати використанням користувачами послуг та ресурсів.

Політика використання ресурсів, що реалізується КЗЗ, стосується:

– користувачів загальнодоступної інформації;

– адміністратора безпеки та користувачів, яким надано повноваження щодо управління АС;

– файлової системи;

– системного та функціонального програмного забезпечення;

– технологічної інформації щодо управління АС;

– окремих периферійних пристроїв (принтерів, накопичувачів інформації і т.ін.);

– обчислювальних ресурсів АС і передбачає можливість встановлення обмежень на їх використання.

Обмеження щодо використання окремим користувачем та/або процесом обсягів обчислювальних ресурсів АС або кількості об’єктів встановлюються адміністратором безпеки або користувачами, яким надано повноваження щодо управління АС. Запити на зміну встановлених обмежень повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від зазначених користувачів.

Спроби користувачів перевищити встановлені обмеження на використання ресурсів повинні реєструватися в системному журналі.

5.3.7 Відновлення після збоїв

КЗЗ повинен реалізувати рівень ДВ-1.

Політика відновлення після збоїв, що реалізується КЗЗ, стосується:

– системного та функціонального програмного забезпечення;

– засобів захисту інформації та засобів управління КСЗІ;

– засобів адміністрування та управління обчислювальною системою АС – і гарантує повернення АС у відомий захищений стан після відмов або переривання обслуговування, спричинених помилковими діями користувачів, неврахованою функціональною недостатністю програмного та апаратного забезпечення (наприклад, можливою наявністю не виявлених під час проектування незадекларованих функцій), іншими непередбачуваними ситуаціями.

Політика відновлення, яка реалізується КЗЗ, повинна визначати множину типів відмов веб-сторінки і переривань обслуговування, після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Для кожної з відмов повинні бути чітко визначені і задокументовані рівні відмов, у разі перевищення яких необхідна повторна інсталяція веб-сторінки.

Після відмови веб-сторінки або переривання обслуговування, КЗЗ повинен перевести веб-сторінку до стану, з якого повернути її в режим нормального функціонування може тільки адміністратор безпеки і користувачі, які мають повноваження щодо управління АС. Для кожного з них повинна бути визначена множина допустимих виконуваних ними операцій з метою повернення АС у відомий захищений стан.

Повернення АС з режиму, що визначається погіршеними характеристиками обслуговування, в режим нормального функціонування повинно здійснюватися за допомогою ручних (не автоматизованих) процедур.

5.3.8 Реєстрація

КЗЗ повинен реалізувати рівень НР-2. Послуга дозволяє контролювати небезпечні відповідно до політики безпеки веб-сторінки дії користувачів всіх категорій із захищеними об’єктами.

Політика реєстрації стосується:

– користувачів усіх категорій;

– публічної інформації веб-сторінки;

– системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження веб-сторінки;

– створеної в процесі супроводження веб-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС.

КЗЗ повинен забезпечувати реєстрацію всіх подій, які мають безпосереднє відношення до безпеки. До них відносяться наступні класи подій:

– вхід/вихід або намагання входу/виходу в/із системи користувачами будь-яких категорій;

– реєстрація та видалення або намагання реєстрації та видалення користувачів будь-якої категорії в системі;

– зміна атрибутів доступу користувачем будь-якої категорії та дії, що призвели до цього;

– отримання або намагання отримання доступу користувачем будь-якої категорії до будь-яких захищених процесів і об'єктів АС;

– створення користувачем будь-якої категорії твердих копій та виведення їх на друкуючі пристрої;

– модифікація або спроби модифікації захищених процесів і об'єктів АС, у тому числі факти та спроби порушення цілісності КЗЗ;

– спроби використання обчислювальних ресурсів АС з перевищенням встановлених квот;

– інші події, обов’язковість реєстрації яких передбачена політикою реалізації окремих послуг безпеки інформації.

КЗЗ повинен надавати можливість визначення переліку реєстраційних подій виключно адміністратору безпеки.

Реєстрація всіх подій, що мають безпосереднє відношення до безпеки, здійснюється в журналі реєстрації, який повинен містити інформацію стосовно дати, часу, місця, типу і наслідків зареєстрованої події (успішність/неуспішність), ім’я (IP-адресу) та/або ідентифікатор причетного до цієї події користувача.

Реєстраційна інформація повинна бути достатньою для однозначної ідентифікації користувача, процесу і/або об'єкта, що мали відношення до кожної зареєстрованої події.

КЗЗ повинен мати механізми захисту для гарантування безпечної передачі інформації журналу реєстрації на віддалену робочу станцію адміністратора безпеки веб-сторінки (для технології Т2).

Адміністратор безпеки і користувачі, яким надано повноваження щодо управління АС, повинні мати засоби перегляду та аналізу журналу реєстрації, а КЗЗ повинен забезпечувати захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування.

5.3.9 Ідентифікація і автентифікація

КЗЗ повинен реалізувати рівень НИ-2. Ідентифікація і автентифікація дозволяють КЗЗ визначити і перевірити особу суб’єкта, що намагається одержати доступ до захищених об’єктів веб-сторінки.

Політика ідентифікації і автентифікації стосується:

– всіх користувачів веб-сторінки, які намагаються одержати доступ до системного та функціонального програмного забезпечення, що використовується для актуалізації, захисту публічної інформації та супроводження веб-сторінки;

– створеної в процесі супроводження веб-сторінки технологічної інформації КСЗІ та технологічної інформації щодо управління АС;

– задіяного для цього периферійного обладнання.

КЗЗ повинен однозначно ідентифікувати категорії користувачів веб-сторінки і за атрибутами кожної з цих категорій визначати послуги, що їм доступні. Ідентифікація здійснюється на підставі особистого імені та/або IP-адреси користувача.

КЗЗ повинен автентифікувати адміністратора веб-сторінки, співробітників СЗІ та користувачів, які мають повноваження щодо управління АС, з використанням захищеного механізму на підставі особистого пароля.

Автентифікація користувачів, що мають виключне право доступу тільки до публічної інформації, не здійснюється. Дозвіл на виконання будь-яких дій з

інформацією та обладнанням веб-сторінки, що контролюються КЗЗ, надається користувачу тільки після успішного завершення процедур ідентифікації та/або автентифікації його КЗЗ відповідно до категорії користувача.

КЗЗ повинен забезпечувати захист даних автентифікації від несанкціонованого доступу, модифікації або руйнування.

5.3.10 Ідентифікація і автентифікація при обміні

КЗЗ повинен реалізувати рівень НВ-1. Ця послуга дозволяє у разі використання технології Т2 компонентам КЗЗ веб-сервера і віддаленої робочої станції здійснити взаємну ідентифікацію, перш ніж розпочати взаємодію.

Послуга ідентифікації і автентифікації при обміні стосується адміністратора безпеки та користувачів, яким надані повноваження щодо супроводження веб-сторінки, технологічної інформації КСЗІ.

КЗЗ повинен надавати доступ до процесів, що забезпечують ініціалізацію обміну даними, тільки адміністратору безпеки і користувачам, яким надано повноваження щодо супроводження веб-сторінки.

Обмін інформацією між компонентами КЗЗ повинен здійснюватися тільки після ідентифікації і автентифікації КЗЗ-відправником КЗЗ-отримувача інформації. Результати процедури ідентифікації і автентифікації є дійсними протягом всього сеансу обміну (незалежно від кількості об’єктів, що експортуються) і втрачають свою силу після його закінчення. Процедура ідентифікації і автентифікація компонентів КЗЗ повинна здійснюватися на підставі їхніх імен, IP-адрес і паролів. Підтвердження ідентичності має виконуватися на підставі затвердженого в АС протоколу автентифікації.

5.3.11 Достовірний канал

КЗЗ повинен реалізувати рівень НК-1. Ця послуга повинна гарантувати користувачу будь-якої категорії можливість безпосередньої взаємодії з КЗЗ, а також те, що ніяка взаємодія користувача з АС не може бути модифікованою іншим користувачем або процесом. Послуга визначає вимоги до механізму встановлення достовірного зв'язку між користувачем і КЗЗ.

Політика достовірного каналу стосується користувачів усіх категорій та компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ.

Достовірний канал повинен використовуватися для початкової ідентифікації і автентифікації. Зв'язок з використанням даного каналу повинен ініціюватися виключно користувачем.

5.3.12 Розподіл обов’язків

КЗЗ повинен реалізувати рівень НО-1. Ця послуга дозволяє розмежувати повноваження користувачів, визначивши категорії користувачів з певними і притаманними для кожної з категорій функціями (ролі). Послуга призначена для зменшення потенційних збитків від навмисних або помилкових дій користувачів і обмеження авторитарності керування АС.

Політика розподілу обов’язків, що реалізується КЗЗ, стосується користувачів усіх категорій і повинна визначати щонайменше такі ролі:

– адміністратора безпеки;

– користувачів, яким надано право доступу до певних видів інформації (публічної, технологічної, системного та функціонального ПЗ).

Кількість користувачів, які мають доступ до технологічної інформації та системного і функціонального ПЗ повинна бути мінімізована, щоб обмежити їх коло тільки тими, кому необхідний такий доступ для виконання функціональних обов'язків, що передбачаються експлуатаційною та розпорядчою документацією на веб-сторінку.

Адміністратору безпеки дозволяється доступ до всієї інформації веб-сторінки. У разі необхідності його роль може дублюватися уповноваженим співробітником СЗІ. Повноваження всіх інших користувачів щодо доступу до інформації надаються їм адміністратором безпеки.

КЗЗ повинен присвоїти користувачу атрибути, якими однозначно характеризується надана йому роль. Користувач може виступати в певній ролі тільки після того, як він виконає дії, що підтверджують прийняття ним цієї ролі.

5.3.13 Цілісність комплексу засобів захисту

КЗЗ повинен реалізувати рівень НЦ-1. Ця послуга визначає міру здатності КЗЗ веб-сторінки захищати себе і гарантувати свою спроможність керувати захищеними об’єктами. Політика цілісності КЗЗ повинна визначати склад КЗЗ, механізми контролю цілісності його компонентів та порядок їх використання.

Політика цілісності КЗЗ стосується: адміністратора безпеки; окремих компонентів системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ; засобів захисту інформації, а також технологічної інформації КСЗІ і забезпечує взаємодію зазначених об’єктів.

Політика реалізації послуги повинна гарантувати, що всі послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ. Якщо існують обмеження, недотримання яких може призвести до надання послуг в обхід інтерфейсу КЗЗ і порушення цілісності КЗЗ, то такі обмеження повинні бути описані і задокументовані. До користувачів має бути доведено порядок їх роботи з дотриманням цих обмежень, а КЗЗ повинен надавати адміністратору можливість здійснення контролю за цим порядком.

КЗЗ повинен повідомляти адміністратора безпеки про порушення цілісності будь-якого компонента КЗЗ. веб-сторінка під час цього має бути переведена до стану, в якому доступ до неї користувачів, крім адміністратора безпеки, заборонено. Повернення до нормального режиму функціонування може бути здійснено тільки адміністратором після відновлення відповідності цього компонента еталону.

5.3.14 Самотестування

КЗЗ повинен реалізувати рівень НТ-1. Самотестування дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій захисту веб-сторінки.

Політика самотестування поширюється на адміністратора безпеки, компоненти системного та функціонального програмного забезпечення, які задіяні для реалізації механізмів КЗЗ, засоби захисту інформації.

До складу КЗЗ повинна входити множина тестових процедур, яка враховує особливості функціонування компонентів конкретної веб-сторінки і достатня для оцінки правильності виконання всіх критичних для безпеки публічної та технологічної інформації КСЗІ функцій, а сам КЗЗ повинен бути здатним контролювати їх виконання.

КЗЗ повинен забезпечувати виконання тестів за запитом адміністратора безпеки. У разі некоректного виконання якогось із тестів КЗЗ повинен перевести АС до стану, коли забороняється надання користувачам доступу до веб-сторінки, або до стану, коли забороняється надання доступу до інформації з використанням функцій, для яких тест не було виконано. Повернути АС до нормального функціонування може тільки адміністратор безпеки після відновлення працездатності КЗЗ і повторного виконання повного набору тестів.

КЗЗ повинен забезпечувати відповідність набору тестів (неможливість будь-якої модифікації) версії КЗЗ. Зміна тестів можлива лише у процесі інсталяції нової версії КЗЗ.

5.4 Опис методики формальної специфікації комплексної системи захисту інформації

Методика формального проектування включає в себе опис інформаційно-телекомунікаційної системи та вимоги політики безпеки інформації.

Формальна модель будується поетапно по наступним крокам:

1) Визначення та відображення комплексу технічних засобів у вигляді діаграм компонентів;

2) відображення зовнішніх фізичних інтерфейсів комплексу технічних засобів у вигляді вузлів компонента;

3) визначення взаємозв’язків між вузлами компонентів комплексу технічних засобів

4) реалізація з’єднань необхідних, наданих або ж комплексних інтерфейсів комплексу технічних засобів;

5) визначення та відображення програмних компонентів, що реалізують технологію обробки інформації або комплекс заходів захисту;

6) відображення зовнішніх інтерфейсів програмних компонентів у вигляді вузлів компонента;

7) визначення взаємозв’язків між вузлами програмних компонентів;

8) реалізація з’єднань необхідних, наданих або ж комплексних інтерфейсів програмних компонентів;

9) визначення та реалізація залежностей між компонентами моделі за допомогою нотації зв’язку залежності;

10) визначення захищених ресурсів (компонентів);

11) надання захищеним ресурсам критичних властивостей;

12) визначення та відображення стереотипу користувачів системи;

13) призначення ролей користувачам системи;

14) призначення користувачам системи правил доступу до захищених ресурсів.

5.5 Часткова формальна модель політики безпеки інформації

В формальній моделі політики безпеки реалізовано наступні компоненти (кожен реалізований компонент виділено червоним кольором):

1) Компонент «Сервер» (рис. 5.1);

2) компонент «Мережева плата» (рис. 5.2);

3) компонент «Операційна система на базі Linux» (рис. 5.3);

4) компонент «Віртуальна файлова система» (рис. 5.4);

5) компонент «Програмний модуль мережі» (рис. 5.5);

6) компонент «Плата контролеру» (рис. 5.6);

7) компонент «Драйвери периферійних пристроїв» (рис. 5.7);

8) компонент «Драйвери доступу до пристроїв» (рис. 5.8);

9) компонент «ПЗ Веб-сервера Apache» (рис. 5.9);

10) компонент «Клієнтська ПЕОМ» (рис. 5.10);

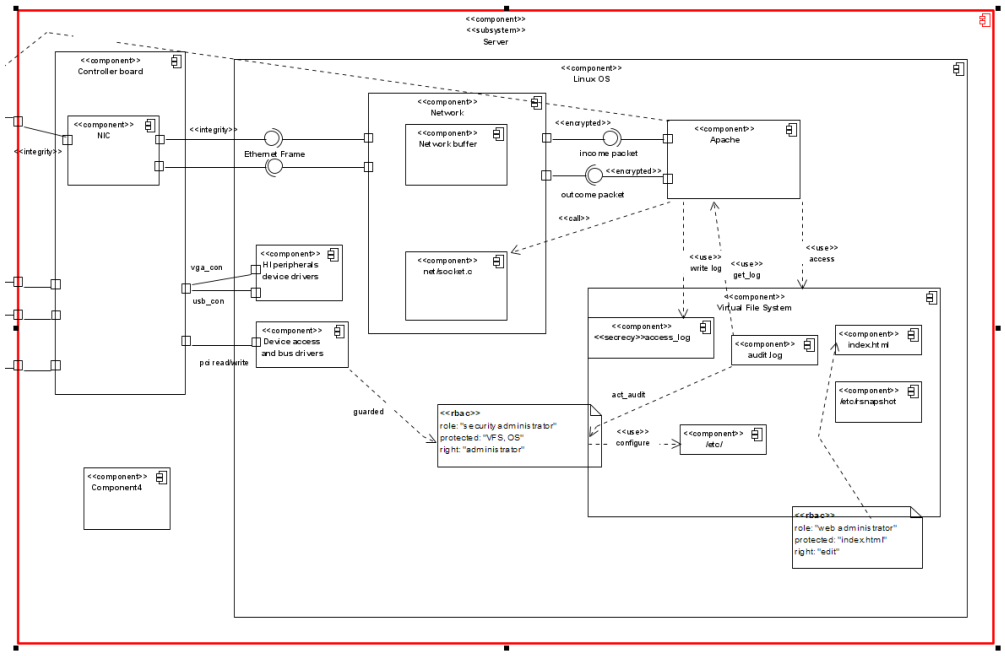

Рисунок 5.1 – Компонент «Сервер»

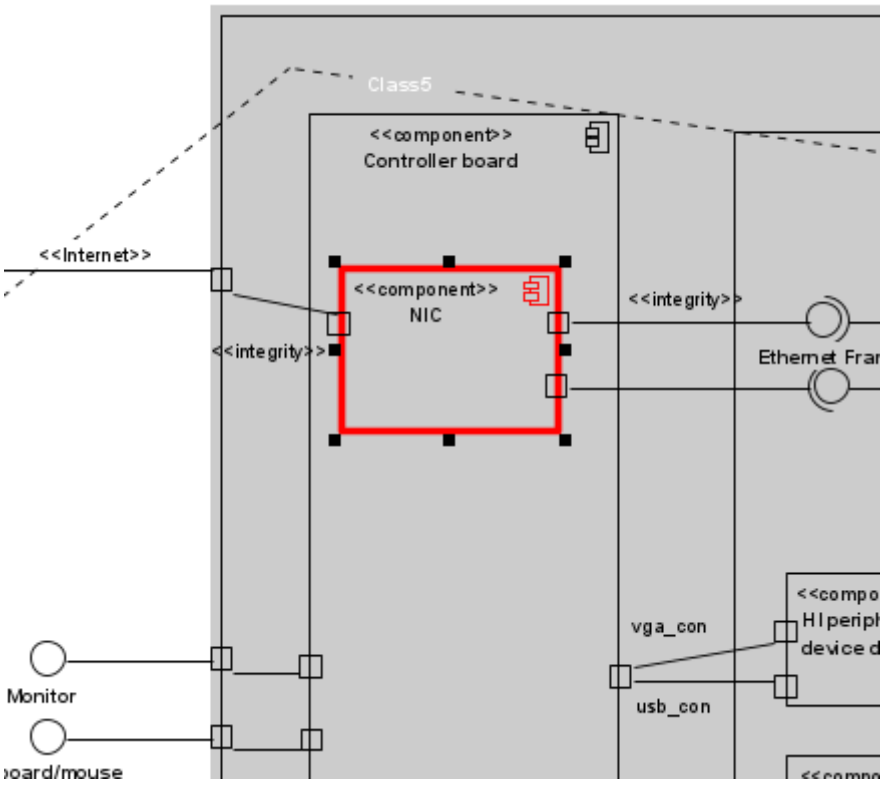

Рисунок 5.2 – Компонент «Мережева плата»

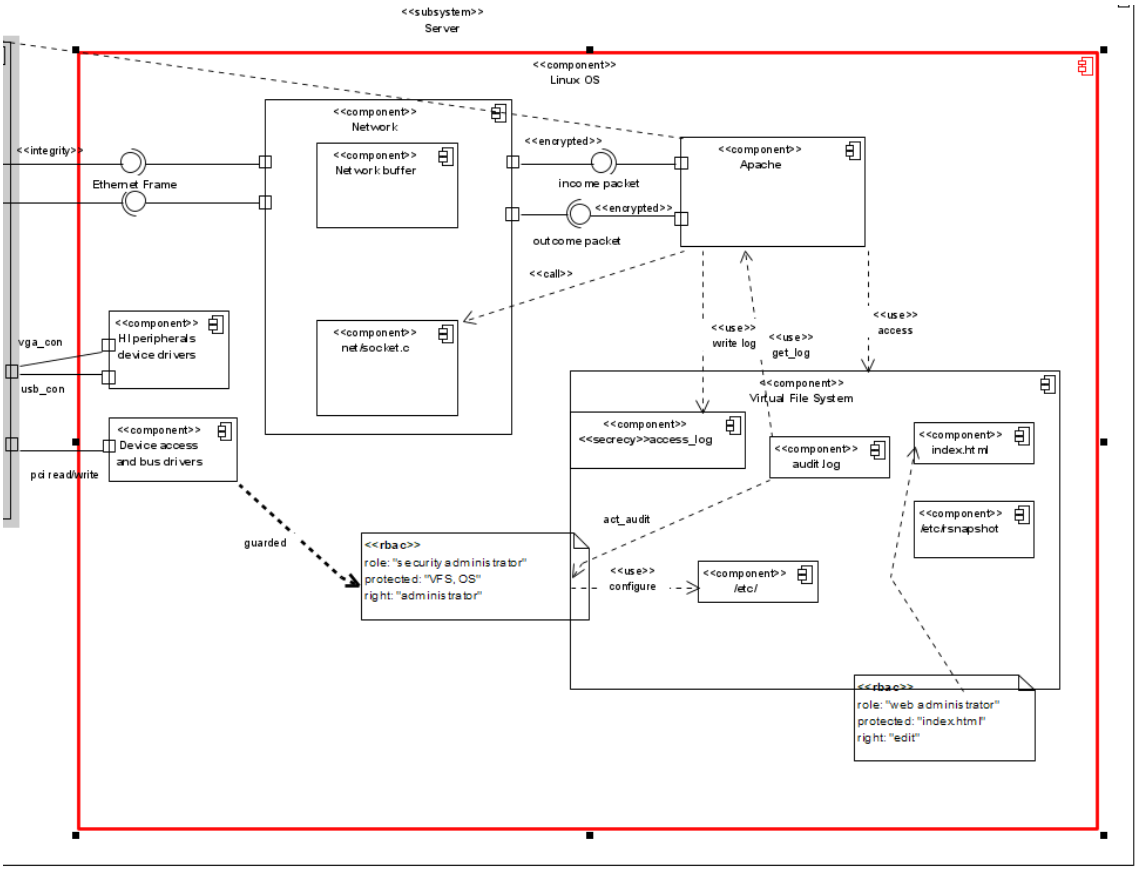

Рисунок 5.3 – Компонент «ОС»

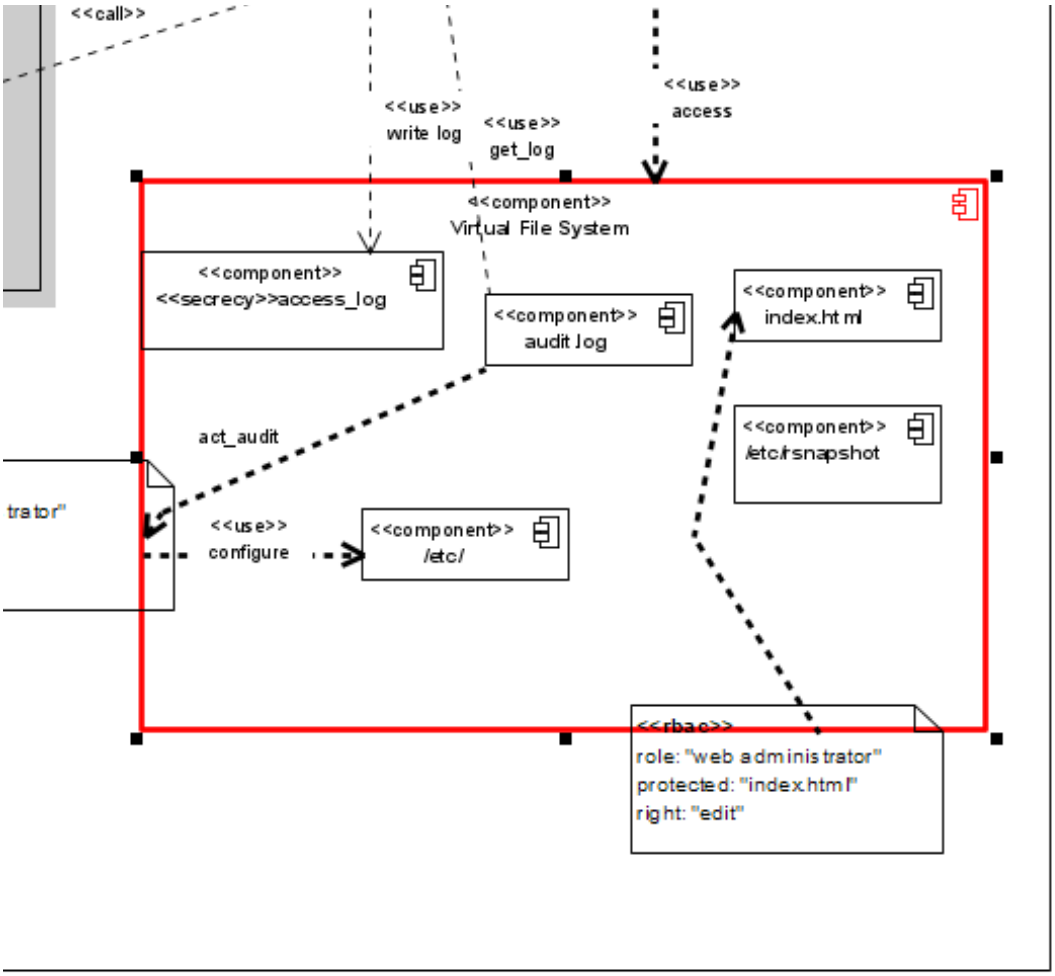

Рисунок 5.4 – Віртуальна файлова система

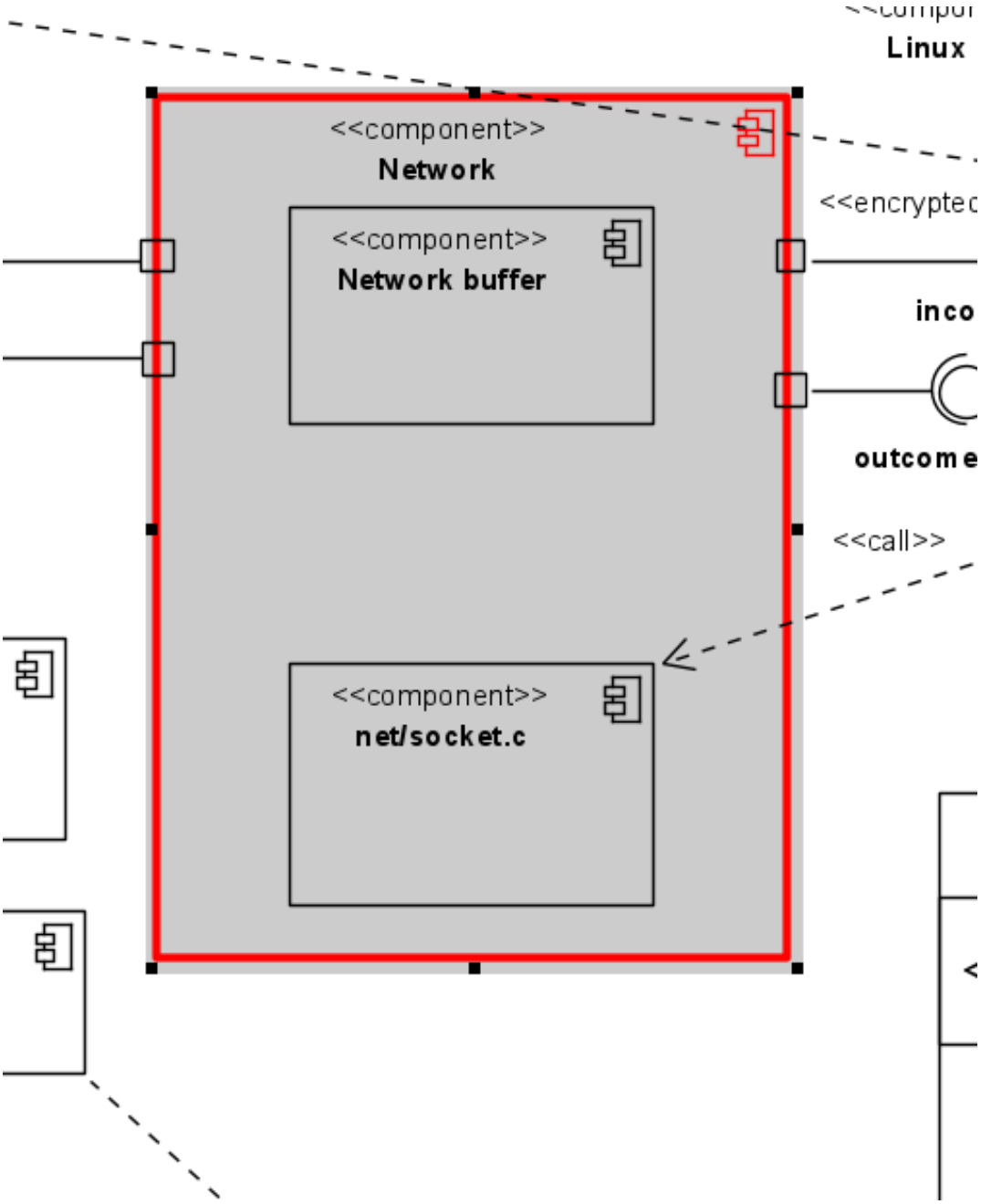

Рисунок 5.5 – Програмний модуль мережі

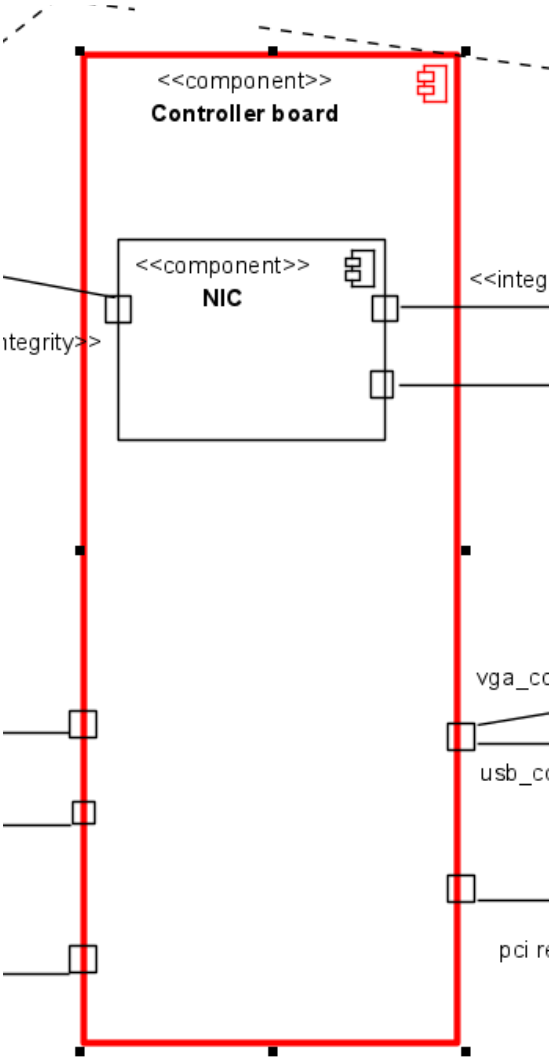

Рисунок 5.6 – Компонент «Плата контролеру»

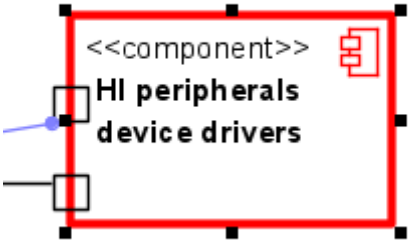

Рисунок 5.7 – Реалізація компоненту «Драйвери периферійних пристроїв»

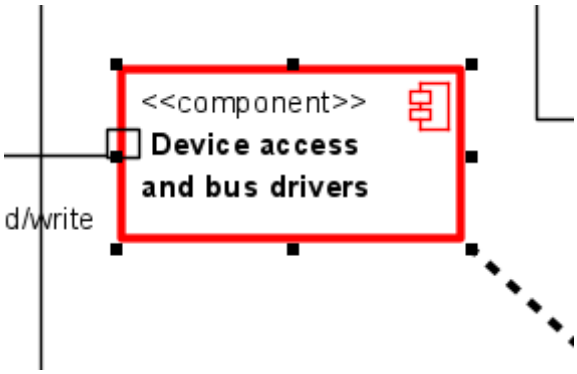

Рисунок 5.8 – Компонент «Драйвери доступу до пристроїв»

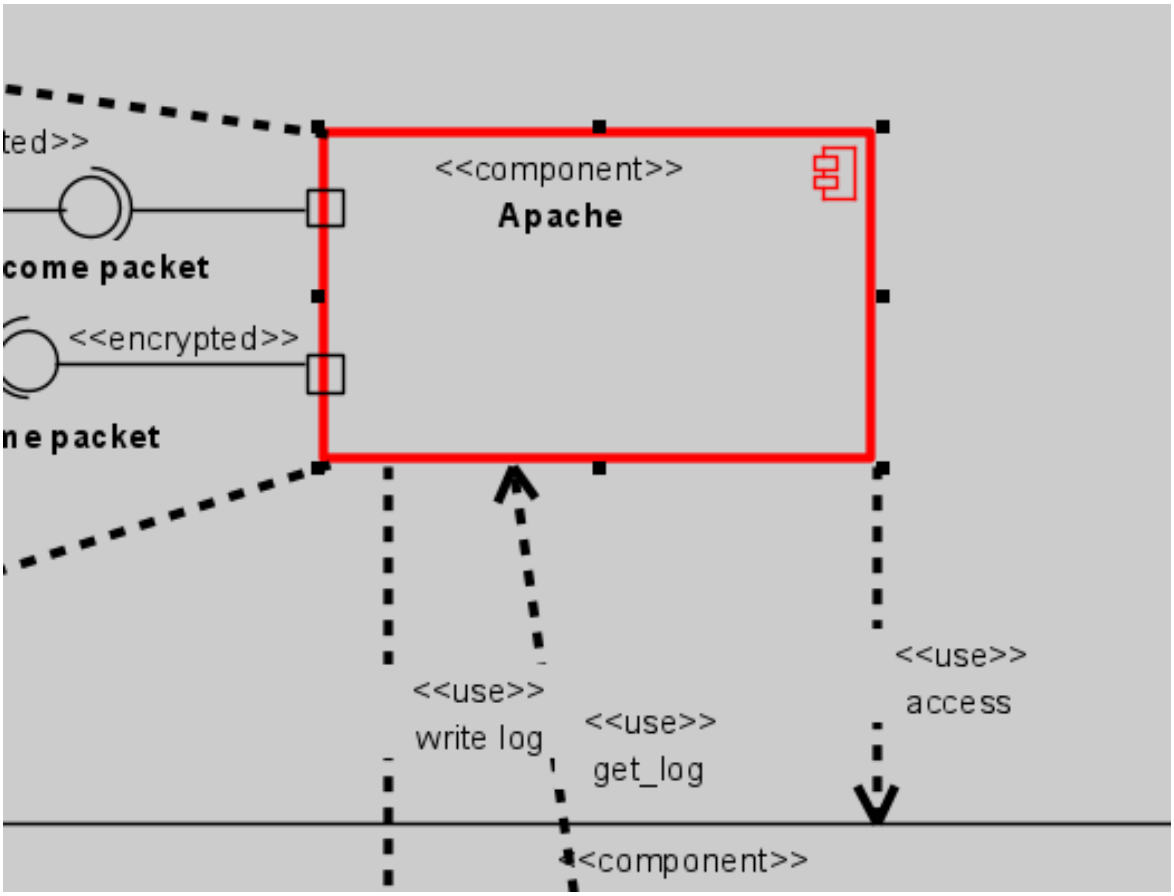

Рисунок 5.9 – компонент «ПЗ Веб-сервера Apache»

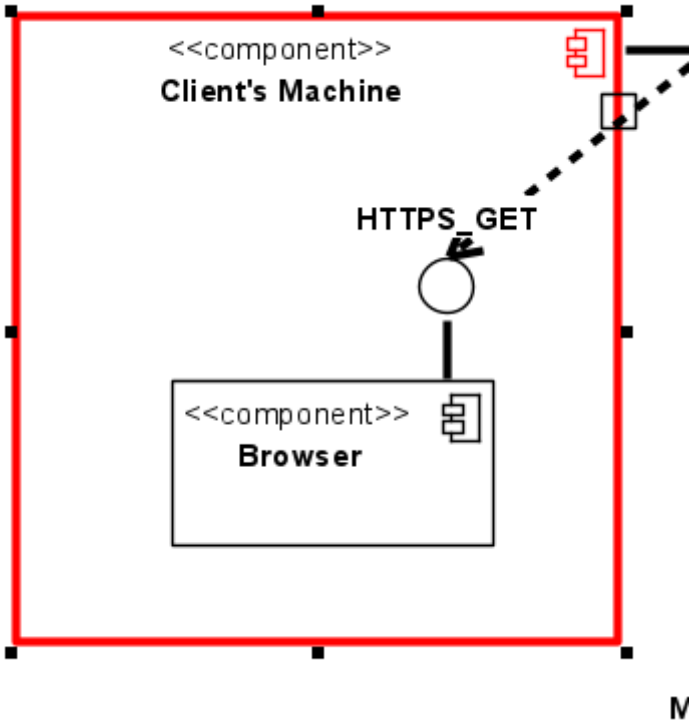

Рисунок 5.10 – Клієнтська ПЕОМ

Реалізація середовища користувачів реалізовано за допомогою стереотипу <<rbac>>, що визначає права та ролі в системі (рис. 5.11 - 5.12).

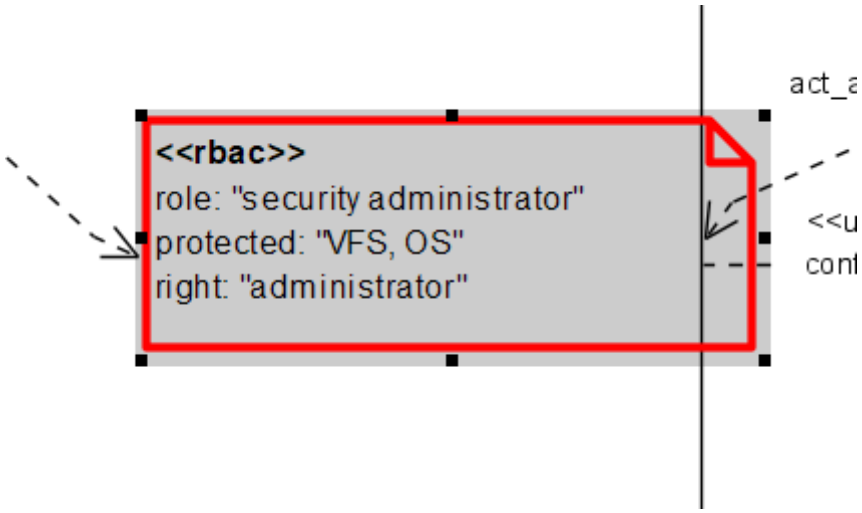

Рисунок 5.11 – Реалізація середовища користувачів (адміністратор безпеки)

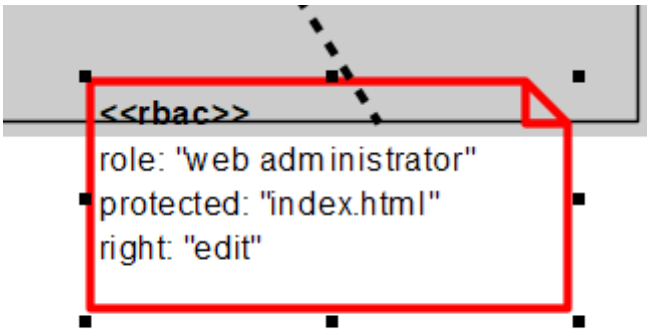

Рисунок 5.12 – Реалізація середовища користувачів (адміністратор веб-сторінки)

На рисунку 5.13 зображено загальні правила використання та розмежування доступу до веб-сторінки.

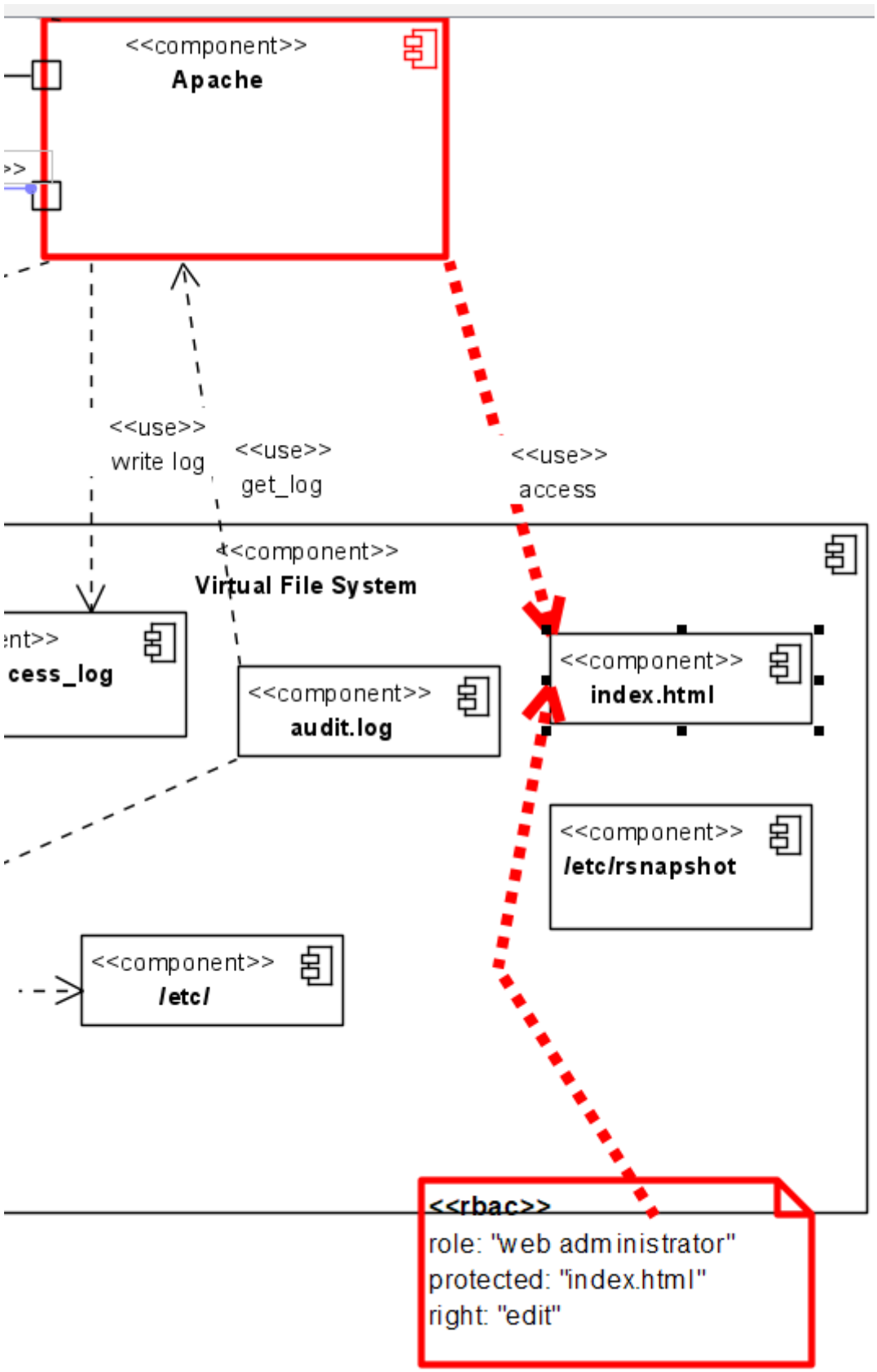

Рисунок 5.13 – ПРД веб-сторінки

Реалізація реєстрації подій зображено на рисунку 5.14.

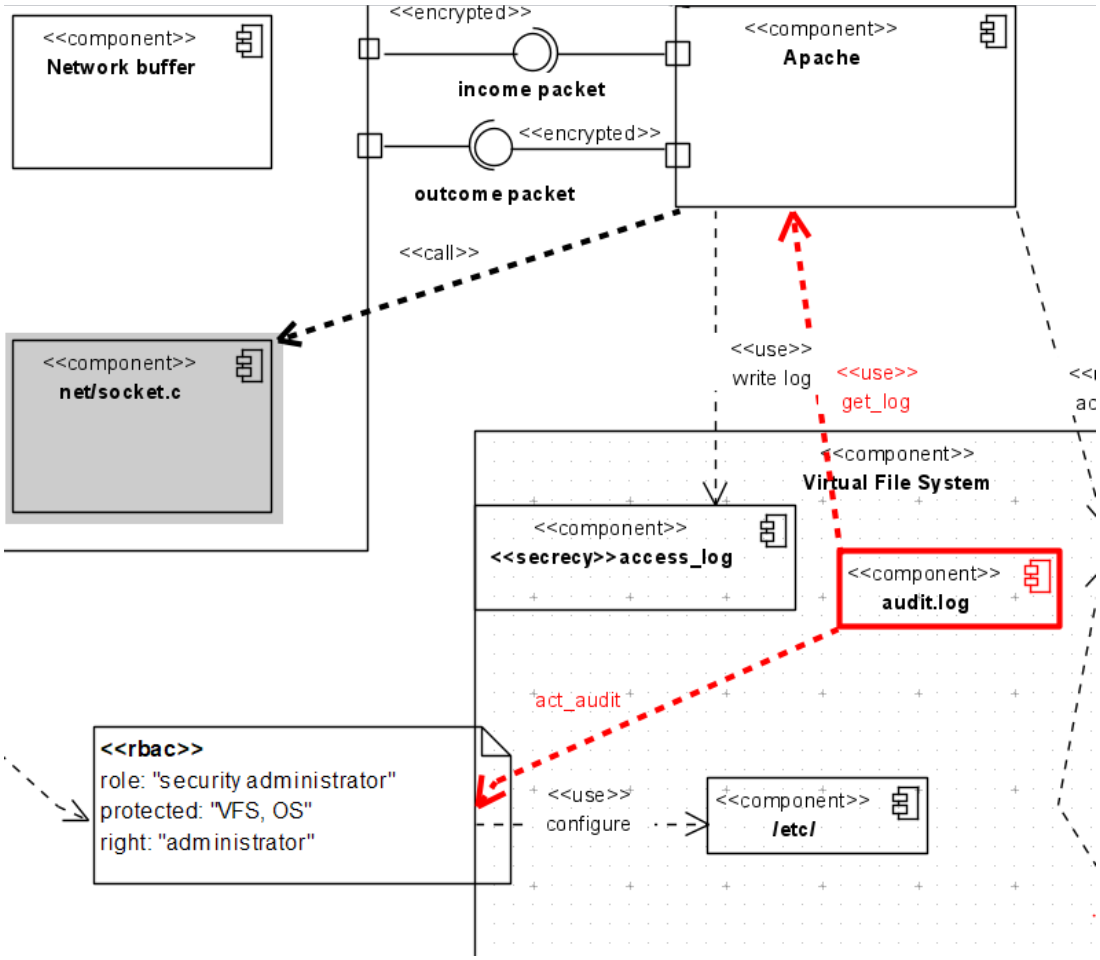

Рисунок 5.14 – Реєстрація подій

Реалізована в рамках демонстрації часткова формальна модель політики інформації ІТС веб-додатку зображена на рисунку 5.15.

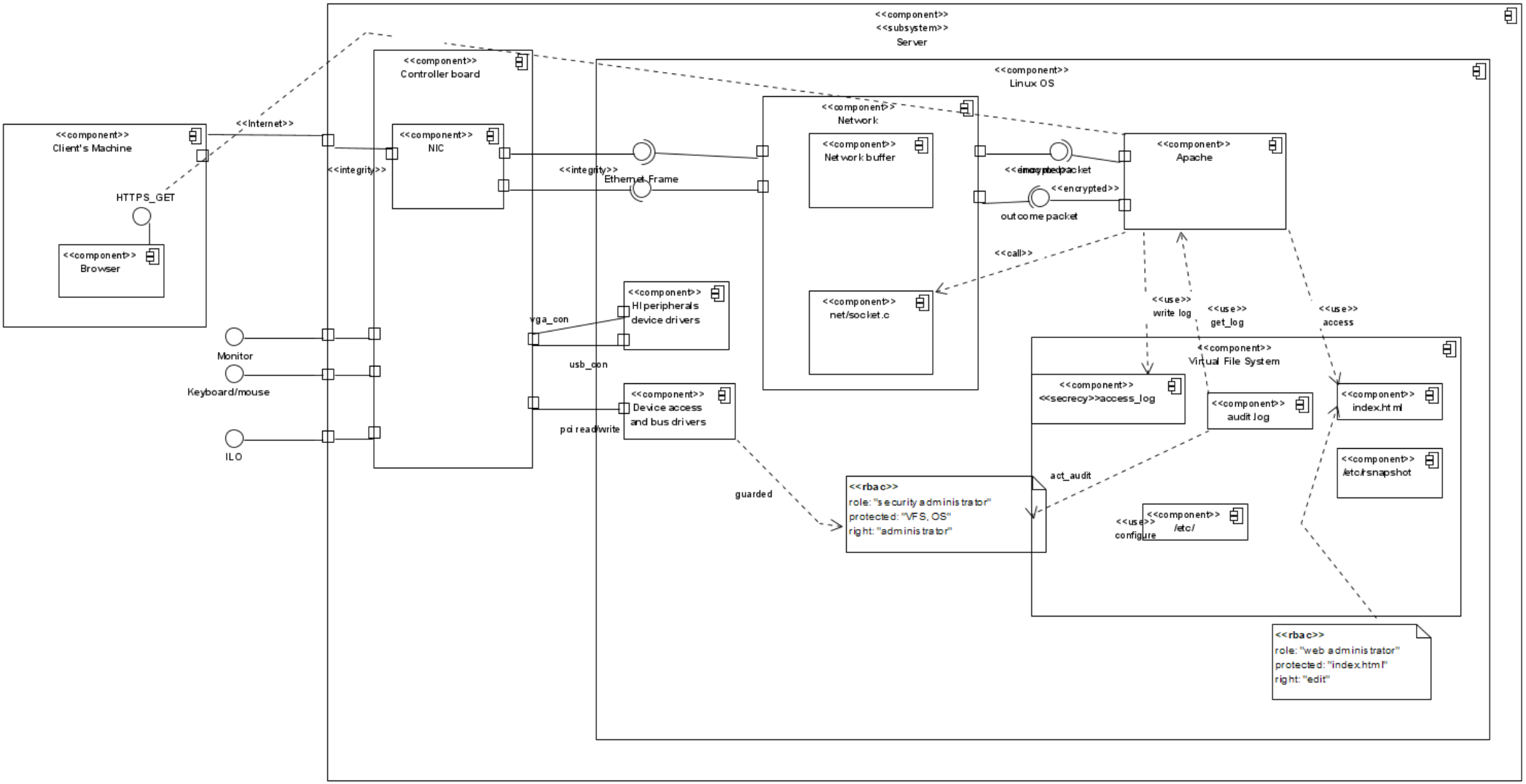

Рисунок 5.15 – Формальна модель (часткова) політики безпеки інформації

ВИСНОВКИ

На сьогоднішній день, Україна на шляху впровадженні низки серії стандартів ISO/IEC 27000, що підтверджують зміни в [2], це доволі довгий процес, а поки побудова систем захисту інформації регламентується нормативними документами в сфері технічного захисту інформації. Комплексна система захисту інформації потребує змін, одним з напрямків діяльності є створення методик для формалізації процесу розробки КСЗІ.

В ході проведення досліджень було проаналізовано методи формального опису ІТС та процесів обробки інформації. Існує тільки один промислово розповсюджений метод опису – UML. Формалізований опис ІТС подається у вигляді діаграми компонентів UML.

Було проаналізовано методи формального моделювання політики безпеки інформації та продемонстровано формальну модель політики безпеки інформації. При моделюванні було використано UMLsec. До недоліків методу можна віднести відсутність критичної властивості інформації – доступності.

В ході проведення досліджень була запропонована методика формального проектування КСЗІ в ІТС, що включає в себе формальний опис ІТС та процесів обробки інформації, формальну модель політики безпеки та показаний процес формування комплексу засобів захисту в ІТС.

У ході подальших досліджень планується розширення профілю UMLsec необхідними параметрами реалізації вимог політики безпеки інформації та функціональних послуг безпеки, такого як критерій доступності інформації. Реалізацію профілю UMLsec доцільно включити до переліку стандартних профілів UML. Для автоматизації процесу розробки планується перехід на платформу розробки Eclipse UML2 Tools для опису моделі програмним кодом Java, результатом компіляції такого коду буде графічна формальна модель.

ПЕРЕЛІК ПОСИЛАНЬ

1. J. J¨urjens Secure Systems Development with UML. Springer – Verlag, 2005.

2. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах»

3. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. Київ: Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, 1999. 61с.

4. НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу. ДСТСЗІ СБ України, Київ, 2003. 24с.

5. НД ТЗІ 2.7-010-09. Методичні вказівки з оцінювання рівня гарантій коректності реалізації функціональних послуг безпеки в засобах захисту інформації від несанкціонованого доступу. Адміністрація Державної служби спеціального зв'язку та захисту інформації України, Київ, 2009. 175с.

6. UML Deployment Diagrams [Електронний ресурс] – https://www.uml-diagrams.org/deployment-diagrams.html

7. OMG Unified Modeling Language (OMG UML), Superstructure Version 2.2 [Електронний ресурс] – https://www.omg.org/spec/UML/2.2/Superstructure/PDF

8. UML Component Diagrams [Електронний ресурс] –https://www.uml-diagrams.org/component-diagrams.html

9. UMLsec. Presenting the Profile. Jan Jurjens [Електронний ресурс] – https://www.omg.org/news/meetings/workshops/DOCsec-2002_Proceedings/01-2_Juergens_UMLsec_Tutorial.pdf

10. Ponder2 Overview [Електронний ресурс] – http://ponder2.net/cgi-bin/moin.cgi/Ponder2Project

11. Гвоздьов Р. Методика формального проектування КСЗІ в ІТС / Р. Гвоздьов, В. Заболотний, А. Бойко // Global Cyber Security Forum : матеріали Першого міжнародного науково-практичного форуму, 14 – 16 листопада 2019 р. – Харьков : ХНУРЭ, 2019. – С. 39–40.

12. Sandrine Duflos, Gladys Diaz, Valérie Gay, Eric Horlait. A Comparative Study of Policy Specification Languages for Secure Distributed Applications [Електронний ресурс] – http://link.springer.com/content/pdf/10.1007%2F3-540-36110-3_16.pdf.

13. Othman Benammar, Hicham Elasri, Mostafa Jebbar and Abderrahim Sekkaki. SECURITY POLICIES MATCHING THROUGH ONTOLOGIES ALIGNMENT TO SUPPORT SECURITY EXPERTS [Електронний ресурс] – https://www.researchgate.net/publication/281995700_Security_policies_matching_through_ontologies_alignment_to_support_security_experts

14. Irem Aktuga, Katsiaryna Naliukab. ConSpec – A Formal Language for Policy Specification [Електронний ресурс] – https://pdf.sciencedirectassets.com/272990/1-s2.0-S1571066108X00064/

15. Zengbang Ma, Yingjie Yang, and Yutong Wang. A Security Policy Description Language for Distributed Policy Self-management [Електронний ресурс] – https://core.ac.uk/download/pdf/147542779.pdf

16. Jun Kong, Dianxiang Xu, Xiaoqin Zeng. Uml-based modeling and analysis of security threats [Електронний ресурс] – https://www.researchgate.net/publication/220344852_Uml-Based_Modeling_and_Analysis_of_Security_Threats

17. Salim Chehida, Mustapha kamel Rahmouni. Security Requirements Analysis of Web Applications using UML [Електронний ресурс] – http://ceur-ws.org/Vol-867/Paper24.pdf

18. David Basin, Jurgen Doser, Torsten Lodderstedt. Model Driven Security: from UML Models to Access Control Infrastructures [Електронний ресурс] – https://people.inf.ethz.ch/basin/pubs/mdac.pdf

19. Denis Hatebur, Maritta Heisel, Jan J¨urjens, Holger Schmidt. Systematic Development of UMLsec DesignModels Based On Security Requirements [Електронний ресурс] – https://www.researchgate.net/publication/317277357_Systematic_Development_of_UMLsec_Design_Models_Based_On_Security_Requirements

20. Holger Schmidt and Jan J¨urjens. UMLsec4UML2 –Adopting UMLsec to

Support UML2∗Using UMLsec4UML2 for the Specification of Architectural Security Patterns [Електронний ресурс] – https://www.researchgate.net/publication /317277523_UMLsec4UML2_-_Adopting_UMLsec_to_Support_UML2

21. Jan Jurjens. UMLsec Presenting the Profile [Електронний ресурс] – https://www.omg.org/news/meetings/workshops/DOCsec-2002_Proceedings/01-2_Juergens_UMLsec_Tutorial.pdf

22. Kevin Twidle, Naranker Dulay, Emil Lupu and Morris Sloman. Ponder2: A Policy System for Autonomous Pervasive Environments [Електронний ресурс] – https://www.researchgate.net/publication/221039848_Ponder2_A_Policy_System_for_Autonomous_Pervasive_Environments

23. Donatas Mažeika and Rimantas Butleris. MBSEsec: Model-Based Systems Engineering Method for Creating Secure Systems [Електронний ресурс] – https://www.mdpi.com/2076-3417/10/7/2574/pdf

24. Peter Barna, Flavius Frasincar, Geert-JanHouben and Richard Vdovjak. Methodologies for Web Information System Design [Електронний ресурс] – https://www.researchgate.net/publication/2917980_Methodologies_for_Web_Information_System_Design

25. Quentin Rouland, Brahim Hamid, Jean-Paul Bodeveix, Mamoun Filali. A Formal Methods Approach to SecurityRequirements Specification and Verification [Електронний ресурс] – https://www.researchgate.net/publication/336965551_A_Formal_Methods_Approach_to_Security_Requirements_Specification_and_Verification

26. Stephen Chong, Joshua Guttman, Anupam Datta, Andrew Myers, Benjamin Pierce, Patrick Schaumont, Tim Sherwood, Nickolai Zeldovich. Report on the NSF Workshop on Formal Methods for Security [Електронний ресурс] – https://arxiv.org/pdf/1608.00678.pdf

27. Antonio F. Gómez Skarmeta. Introduction to Modelling Languages [Електронний ресурс] – http://www.deserec.eu/files/first_workshop /pdf/DESEREC_UMU_Intro_Modelling_Languages_Workshop_2006.pdf

28. Javier Cánovas. A UML Profile for Privacy Enforcement [Електронний ресурс] – https://modeling-languages.com/a-uml-profile-for-privacy-enforcement/

29. Guilherme Ermel, Kleinner Farias, Vinicius Bischoff, Lucian José Gonçales. Supporting the Composition of UML Component Diagrams [Електронний ресурс] – https://www.researchgate.net/publication/329335043_Supporting_the_Composition_of_UML_Component_Diagrams

30. Chris Lüer, David S. Rosenblum. ML Component Diagrams and Software Architecture—Experiences from the WREN Project [Електронний ресурс] – https://www.researchgate.net/publication/2378060_UML_Component_Diagrams_and_Software_Architecture_-_Experiences_from_the_Wren_Project