1. *Сидоров М.О.* Безопасность программного обеспечения авиационных систем. Монографія // *М.О. Сидоров, М.Г. Луцький, Н.С. Кулик*. - Энциклопедия безопасности авиации.- К.: «Техніка», 2008.- 998с.
2. *Пащенко С.В.* Система підтримки справності та надійності авіаційної техніки при її експлуатації за технічним станом // *С.В. Пащенко, М.Ф. Хільченко.* - «Актуальні проблеми розвитку авіаційної техніки»:- Тези доповідей Науково-практична конференція.- НАУ. - м.Київ. – 2009. – С.87.

Поступила 24.01.2011р.

УДК 004.921

Л. Є. Шведова

РОЗРОБКА МЕТОДУ ФОРМУВАННЯ ПОТОЧНИХ ПОВНОВАЖЕНЬ

Реалізація алгоритмів довільних процесів, що пов'язані з управлінням повноважень, передбачає більш однозначну інтерпретацію всіх елементів, які проектується реалізувати у вигляді алгоритмів. Це обумовлює необхідність звузити та конкретизувати основні елементи, які необхідно використовувати. У зв'язку з цим розглянемо ряд обмежень та уточнень завдяки яким стане можливим реалізувати алгоритми, що безпосередньо складають систему *SUP*:

— поетапні обмеження процесу функціонування *SUP*;
— структурні обмеження способу організації *SUP*;
— параметричні обмеження, які визначають певний перелік параметрів;
— функціональні обмеження;
— обмеження діапазонів значень.

Поетапність процесу функціонування обумовлюється тим, що в рамках *IS* встановлюється період циклу функціонування системи. Основною ознакою завершення такого циклу є необхідність забезпечити початкові значення для всіх учасників процесу, який реалізується системою *SUP*. У випадку синхронізації етапу функціонування *SUP* прикладними задачами, що розв'язуються в *IS*, ознаками завершення одного циклу роботи *SUP* є такі події, що відбуваються в прикладній системі:

— завершення розв'язку задач в *IS*;
— виникнення події, що визначається, як ознака завершення циклу функціонування прикладної задачі в *IS*;
— виникнення в прикладній задачі аварійної події.

Перший випадок завершення циклу є найбільш природним і відповідає логіці розв'язку задачі.

Другий випадок являє собою ситуацію, коли в *IS* виникає подія, яка передбачена проектом прикладної задачі і не визначає успішного її завер-

шення, але з точки зору логіки функціонування задачі може визначати ініціацію початку нового циклу функціонування *SUP*.

Аварійною подією, що може виникнути в процесі функціонування прикладної задачі, є подія, яка призводить до неможливості розв'язку прикладної задачі і є непередбачуваною в рамках алгоритмів, що реалізують розв'язок прикладної задачі. Прикладом таких подій може бути зациклювання процесу функціонування прикладної задачі, формування вихідних даних, які є недопустимими, або непередбачуваними логікою реалізації прикладної задачі та інші ситуації, характер яких тісно пов'язаний з особливостями прикладних задач, що розв'язуються в *IS* з використанням *SUP*.

Завершення етапу функціонування *SUP* може синхронізувати системними подіями, до яких відносяться:

— встановленими значеннями системних параметрів, що характеризують *SUP*;
— перехід компонент *SUP* у стан з початковими значеннями параметрів, що їх характеризують;
— технологічними умовами функціонування *SUP*.

Системні параметрі *SUP*, як і параметри будь-якої іншої системи, характеризують останню незалежно від тих чи інших властивостей прикладних задач, які вона може обслуговувати. Такі параметри носять інтегральний характер і не відображають особливості окремих компонент *SUP*. Наприклад, окремого компонентою *SUP* може бути матриця доступу, а параметром, що відображає її особливість, або персональним параметром є її розмір. Зрозуміло, що цей параметр при необхідності може бути змінений, наприклад, якщо виявиться, що матриця не вміщає всіх атрибутів.

Перехід компонент *SUP* в стан з початковими значеннями параметрів, переважно, реалізується у випадках порушень в роботі системи *SUP* та *IS*. Розглянемо більш детально цю причину.

Технологічні умови функціонування *SUP* передбачають проведення контролю роботи та стану *SUP* незалежно від результатів розв'язку прикладних задач. Вони визначаються умовами технічної експлуатації системи в цілому.

Структурні обмеження, які необхідно впровадити в *SUP*, дозволяють фіксувати структуру і, відповідно, функціональні складові системи. Такі обмеження зумовлюють можливість на встановлений період часу визначити допустимі взаємозв'язки між окремими компонентами. Це означає, що у випадку, коли деяка компонента буде ініціювати передачу даних, або ініціацію функціонування деякої іншої компоненти поточна структура *SUP*, не передбачує можливість встановлення такого зв'язку, то в рамках *SUP* буде фіксуватися подія, яка визначається порушенням структури, що інтерпретується як виникнення атаки на систему доступу. В цьому випадку структурні обмеження можуть бути орієнтовані на різні типи зв'язків між компонентами, до яких відносяться:

— обмеження на передачу даних;

— обмеження на зчитування даних;
— обмеження на ініціацію компоненти;
— комбіновані обмеження.

Структурні обмеження, що реалізуються в рамках *SUP*, повинні дотримуватися в межах одного циклу функціонування *IS*, який визначається подією, що полягає у завершенні розв'язку системи задач, яка була ініційована в рамках відповідної структури взаємодії компонент *SUP* та в цілому, в рамках *IS*.

В *SUP* використовується цілий ряд параметрів, що описують можливості *SUP*. До таких параметрів відносять:

— певна сукупність типів повноважень, якими можуть користуватися суб'єкти в рамках поточної версії системи;
— встановлена система пріоритетів надання повноважень, якщо виникла ситуація по запиту одного і того ж повноваження відносно одного і того ж об'єкту у різних суб'єктів однозначно;
— допустима кількість аномалій, що виникають в процесі роботи *SUP* та їх типів;
— певна сукупність компонент *SUP*, які використовуються для її реалізації;
— спосіб управління за запитами, які встановлюють суб'єкти y_i, для отримання відповідних повноважень.

Та чи інша сукупність повноважень в першу чергу визначається особливостями прикладних задач, які розв'язуються в *IS*. Крім найбільш поширених типів повноважень, до яких відносяться: читання даних з x_i, запис даних в x_i, заміна даних в x_i та ін. В *SUP* можуть аналізуватися повноваження, що активізують окремі об'єкти, які можуть інтерпретуватися компонентами *IS*, що по своїй природі є суб'єктами. Наприклад, якщо деяка програма або програмний модуль повинен активізуватися інших суб'єктом y_i, то відповідний суб'єкт y_i повинен, на основі використання повноважень іншим суб'єктом, перетворити його в об'єкт і тоді y_i може скористатися повноваженнями з активізації відповідного об'єкту $x_i(y_i)$. Може виникнути питання, чому б суб'єкту y_i відразу не надати повноважень по зміні статусу відповідного суб'єкта. Це призведе до зниження рівня безпеки *SUP*, оскільки довільний функціональний y_i зможе отримати повноваження, яке в ієрархії повноважень, або в ієрархії типів повноважень, відноситься до рівня, який є не доступним для окремого конкретного суб'єкта y_j. З наведеного виходить, що в *SUP* повинні існувати різні класи y_i, з різними можливостями. Таким чином існування різних типів суб'єктів може визначатися типами повноважень, які використовуються в рамках прикладних систем з *IS*. Якщо в *IS* змінюється тип прикладної задачі, то відповідно можуть перерозподілятися класи типів повноважень.

Система пріоритетів, що використовується при наданні повноважень може змінюватися в залежності від вимог системи прикладних задач. При цьому такі зміни можуть реалізуватися на рівні вибору моделей пріоритетів.

Відомо цілий ряд таких моделей, які використовуються в обчислюваних системах, наприклад, магазинна модель, стикова модель та цілий ряд інших моделей [1].

Виникнення аномалій в системі *SUP* і відповідно в системі *IS*, може свідчити про наступне:

— недостатньо коректно спроектовану структуру системи *IS*, в цілому, або її компонент включаючи і систему *SUP*;
— виникнення атак на систему *IS* через дію на *SUP*;
— некоректні умови використання та обслуговування системи *IS* в цілому і, в першу чергу, системних компонент, однією з яких є *SUP*.

Некоректне проектування структури *IS* розглядати не будемо, лише зазначимо, що це є можливим у випадку формування нової прикладної системи *IS*, при використані тих самих системних засобів.

Вимоги з методів використання системних засобів, якими є *SUP*, формуються в рамках реалізації алгоритмів компонент, що входять в склад *SUP*. Тому їх окремо аналізувати не будемо. Виходячи з наведеного можна стверджувати, що всі аномалії, які виникають в системі можна розглядати як наслідки атак, що здійснюються на систему.

Щоб можна було успішно виявити аномалії та на їх основі визначати атаки, необхідно визначатися зі скінченою сукупністю таких аномалій. Доцільність введення такого обмеження обумовлюється тим, що для різних прикладних задач, що розв'язуються в *IS* характерні різні класи атак, які будуть призводити до аномалій, що характеризуються різними способами.

Система *SUP*, як і будь яка інша система характеризується певними параметрами. В залежності від вибраних методів управління повноваженнями, *SUP* буде характеризуватися різними параметрами. На відміну від традиційного підходу організації, в рамках даного підходу, базове співвідношення, яке описує спосіб визначення взаємовідношень між суб'єктом і об'єктом, описується таким чином:

$$\left[c_i\left(y_i\right)\,\&\,k_1\left(x_i\right)\,\&\,h_i\left(y_i,x_i\right)\,\&\,L_i\right]\rightarrow\left(y_i\rightarrow x_i\right),$$

де L_i – деяка логічна формула, або логічна система, що визначає умову надання повноважень суб'єкту y_i для використання об'єкта x_i. В цьому випадку, виконується умова, що описується співвідношенням:

$$\left(L_i\subset L^A\right)\&\left[\left(x_i,y_i\right)\subset L_i\right],$$

де L^A – система базових логічних співвідношень які описують умови використання значимості суб'єктів c_i, категорії об'єктів k_i та типи повноважень h_i. Функціональні обмеження що враховуються у відповідній *SUP*, визначаються типами повноважень, що описуються елементами $h_i\subset$ H.

Обмеження діапазонів значень окремих компонент визначаються для області значень величин c_i та k_i, які є дискретними. Для кожного значення c_i та k_i з областей C та K формуються окремі співвідношення в рамках системи L^A. Це означає, що в рамках логічної системи яка описує початкові співвідно-

шення між суб'єктами і об'єктами, визначені та описані у вигляді логічних співвідношень всі доступні взаємозв'язки між y_i та x_i на поточний момент t_i.

Приймаючи до уваги вищенаведені обмеження сформулюємо наступний спосіб реалізації функціонального блоку початкового інсталювання системи *SUP*, що орієнтовна на встановлену сукупність прикладних задач. Компонента початкової інсталяції системи *SUP* буде складатися з наступних базових підсистем, які орієнтовані на різні випадки початкових установок основних параметрів *SUP*:

— початкова установка параметрів *SUP* та системи логічних співвідношень для L^A у випадку, коли система *SUP* повинна ініціюватися вперше на визначений комплект прикладних задач (*PIS*);

— корекція та розширення значень параметрів системи *SUP* у випадку, коли до системи додаються нові прикладні задачі чи окремі прикладні задачі виводяться з системи is (*RPS*);

— аварійне відновлення параметрів системи *SUP*, що відбувається у випадку виникнення аварійних ситуацій, які потребують перевантаження системи з врахуванням причин виникнення аварійних ситуацій в *SUP* (*AIS*).

В рамках інформаційної технології необхідно забезпечувати максимально можливу автоматизацію всіх процесів, що пов'язані не тільки з функціонуванням *SUP*, а й з допоміжними процесами, до яких відносяться процеси, які реалізуються в рамках *PIS*, *RPS* та *AIS*.

Автоматизація цих процесів, як і будь-яких інших, є можлива при умові, що останні можуть бути в потрібній мірі формалізовано описані. Чим вищий рівень формалізації такого опису, тим в більший мірі можна автоматизувати відповідні процеси. Це пов'язується тим, що формалізовані описи в більший мірі піддаються опису алгоритмами, які в тій чи інший мірі потребують втручання зовнішніх факторів, якими можуть бути інші алгоритми, чи фахівці, що реалізують, або вимушені приймати участь у відповідних алгоритмічних процедурах. В цьому випадку рівень формалізації визначається можливістю забезпечити ту чи іншу повноту, або адекватність опису компонент та відповідних перетворень, які передбачається описувати у вигляді елементів алгоритмів та фрагментів алгоритмів, що окремі перетворення реалізують. Ця обставина є відомою в різних областях абстрактних наук і в першу чергу в теоретичних науках, наприклад, математичній фізиці [3]. На відміну від теоретичних галузей науки, в прикладних науках не завжди існує можливість узагальнити уявлення про ті чи інші фактори, які необхідно включати у відповідні алгоритмі, що дозволяє забезпечити необхідний рівень формалізації відповідної компоненти. В прикладних науках для розв'язку відповідної проблеми використовуються підходи, що полягають у введені текстових описів компонент чи перетворень які не піддаються необхідному рівню формалізації. Це обумовлює необхідність вводити уявлення про інформаційні текстові описи, або представлення окремих компонент системи [4]. Інший підхід до розв'язку цієї проблеми полягає у

використанні лінгвістичних змінних, які в подальшому інтерпретуються в рамках уявлень про розмиту математику, в якій найбільш розробленими та дослідженими є розмита арифметика та розмита логіка [5]. В рамках одного підходу будемо використовувати методи, що ґрунтуються на використані інформаційних компонент. На відміну від другого підходу, в цьому випадку автоматизація процесів аналізу та перетворень реалізується на основі введення уявлень про семантичні параметри, які можна оцінювати їх числовими чи логічними значеннями, що визначаються на основі інтерпретації останніх, та приймається при їх введені. Останнє дозволяє реалізувати аналіз на перетворення відповідних числових та логічних величин в рамках відповідних алгоритмів. Це, в свою чергу, забезпечує можливість автоматизації відповідних процесів. Очевидно, що автоматизація не може бути реалізована в повній мірі по відношенню до процесів, які пов'язані з веденням початкових даних, що необхідні для ініціалізації системи *SUP*. Завдяки підходу який використовує уявлення про інформаційні текстові компоненти, або дані така автоматизація може бути реалізована в значно більшій мірі. Ця можливість виявляється за рахунок того, що у випадку використання представлення вхідних даних у вигляді інформаційних текстових описів, які складаються у зручному вигляді для користувача, засоби аналізу та перетворень текстових описів нормалізують відповідні тексти та визначають семантичні параметри окремих елементів текстових описів, числові значення цих параметрів і здійснюють їх попередню перевірку на предмет виявлення у вхідних даних колізій різного характеру, які є проявом їх некоректності.

Вхідні дані, які вводяться в *SUP* в процесі її інсталяції, представляють собою наступне:

— величина категорій об'єктів, які вводяться на початковій стадії функціонування *SUP*, ($x_i \in X$);

— величина значимостей суб'єктів ($y_i \in Y$);

— початкова система правил по визначенню можливості надання повноваження ($L_p^A \subset L^A$);

— система виводу нових правил ($L_N^A \subset L^A$) надання повноважень $y_i \rightarrow x_i$,

— додаткові компоненти *SUP*, які визначають можливість функціонування *SUP*.

При початковій ініціалізації *SUP* необхідно задавати не тільки окремі значення параметрів c_i та k_i, а й масштаб вимірювання відповідних величин. Як неважко зрозуміти, що c_i та k_i повинні вимірюватися однією і тією ж величиною. Прийнято їх визначити, як безрозмірні величини. Очевидно, що на множині X та Y не задається метрика, оскільки співвідношення між y_i та x_i визначається за допомагаю логічних співвідношень функцій, які формуються на основі системи L^A та системи їх перетворень Ξ. Система правил L_p^A та Ξ являють собою певні розширення відповідних систем класичної логіки [6]. Необхідність таких розширень зумовлюється тим, що предметна область

задач, що розв'язується в рамках *IS*, може вимагати специфічну інтерпретацію класичних логічних функцій або виконання тих чи інших обмежень, при здійсненні тих чи інших перетвореннях. Іншими розширеннями правил перетворень можуть бути операції порівняння, операції визначення нерівностей і в багатьох випадках – операції додавання та віднімання. Для визначеності, в рамках реалізації системи Ξ та L_p^A будемо використовувати всі можливі додаткові інтерпретації окремих логічних функцій та операції рівності, нерівності.

Оскільки, на множинах C і K використовується специфічна операція встановлення залежностей між x_i і y_i, яка являє собою певну логічну функцію, то визначення впорядкованості на множині $M = C \cup K$ буде носити частковий характер. Це означає, що впорядкованість буде задаватися функціями рівності і нерівності $(c_i = k_i) \cup (c_i \neq k_i)$. Інтерпретація цих операторів в даному випадку буде такою. Знак рівності визначає факт допустимості надання повноважень суб'єкту y_i по відношенню до об'єкта x_i. Знак нерівності означає, що надання повноважень суб'єкту y_i по відношенню до об'єкта x_i є недопустимим. Формально, цей фактор описується співвідношенням:

$$\left[\left[c_i\left(y_i\right) = k_i\left(x_i\right)\right] \to \left(y_i \to x_i\right)\right] \cup \left[\left[c_i\left(y_i\right) \neq k_i\left(x_i\right)\right] \to \neg\left(y_i \to x_i\right)\right].$$

Розглянемо, чи існує можливість співставляти c_i та c_j, k_i та k_j, а також c_i та k_j між собою крім того, розглянемо можливість надання часткових повноважень суб'єкту по відношенню до об'єкта, що полягають у помноженнях лише по вибраних типах взаємодії y_i з x_i, або по одному типу взаємодії та умові, що всі інші типи взаємодії або типи повноважень є заборонені. Така ситуація є досить поширеною, оскільки, дозволеним типам повноважень може бути тільки читання даних суб'єктом y_i з об'єкта x_i.

При реалізації традиційних підходів для побудови *SUP*, переважно використовуються оператори впорядкування чисел, наприклад , «>» і «≤», а їх інтерпретація переноситься на інтерпретацію, яка прийнята в теорії чисел і визначає одну величину більшого, або меншого від іншого і т.д. В рамках даного підходу, для реалізації порівнянь окремих величин значимості y_i та величин категорій, використовується система визначення пріоритетів. Як уже відмічалось система визначення пріоритетів може ґрунтуватися на основі використання механізмів або алгоритмів, які використовують певні фактори, що є третіми по відношенню до величин, які порівнюються. Наприклад, таким фактором може бути величина поточної кількості активізації суб'єкта, яка у порівнянні з кількістю активізацій іншого суб'єкта є меншою, або більшою. Формально, це можна записати у вигляді співвідношення:

$$\left[A_i^M\left(y_i\right) \supset A_j^M\left(y_j\right)\right] \to \left[c_i\left(y_i\right) > c_j\left(y_j\right)\right],$$

де A_i^M – середнє значення активності y_i за вибрану величину поточного періоду Δt_i функціонування *SUP*. При такому визначенні взаємозалежностей

між c_i і c_j, на відміну від традиційних підходів, остання з часом може змінюватися. Наприклад, якщо для традиційних підходів впорядкованість множини значень c_i та k_i є фіксованою на всьому періоді функціонування *SUP*, то в рамках даного підходу залежність між величинами c_i і c_j, через деякий час може помінятися, що формально запишеться у вигляді:

$$F_i\left\{\Delta T_i, \left[c_i\left(y_i\right) > c_j\left(y_j\right)\right]\right\} \to \left[c_i\left(y_i\right) < c_j\left(y_j\right)\right],$$

де F – деяка функція, яка визначає спосіб зміни величин значення c_i і c_j на протязі часу ΔT_i. В рамках наведеного вище прикладу співвідношення між елементами $A_i^M\left(c_i\right)$ і $A_j^M\left(c_j\right)$ може протягом інтервалу часу ΔT_i змінитися на протилежне, що і призведе до зміни міри значимості $c_i\left(y_i\right)$ та $c_j\left(y_j\right)$. Аналогічний підхід можна використати для порядкування множини значень k_i. Така можливість ґрунтується на тому, що різні y_i можуть активізувати різні x_i з різними мірами активності $A_i^M\left(x_i\right)$. Таким чином, впорядкованість на множині $k_i \in K$ визначається характером активності елементів $c_i \in C$ і в цьому сенсі є похідною. При цьому зміна взаємозалежностей між $k_i\left(x_i\right)$ та $k_j\left(x_j\right)$ не залежить від зміни активності c_i чи c_j. Якщо активності $A_i^M\left(c_i\right)$ чи $A_j^M\left(x_j\right)$ визначати, як величини часткові, які характеризують окремі типи повноважень наприклад, читання даних W^R, чи запис даних W^V в об'єкт x_i, то впорядкованість c_i в множині C може відрізнятися від загальної впорядкованості.

1. *Касьянов В. А.* Субъективный анализ / *В. А. Касьянов* // Монография. – К. : НАУ, 2007.
2. *Столинг В.* Основы защиты ситей. Приложения и стандарты / *В. Столинг*. – М. : Издательский дом «Вильямс», 2002.
3. *Павленко Ю. Г.* Лекции по теоритической механики / *Ю. Г. Павленко*. – М. : ФИЗМАГЛИТ, 2002.
4. *Афанасьева О.Ю.* Методы семантичних перетворень в стеганосистемах / *О.Ю. Афанасьева* / Зб. наук. праць «Моделювання та інформаційні технології» (ІПМЕ НАН України). – К., 2010. – Вип. 56 – С. 188–196.
5. *Чернявская В.Е.* Лингвистика текста: Поликодовость, интертекстуальность, интердискурсивность / *В. Е. Чернявская*. – М. : Книжный дом «ЛИБРОКОМ», 2009.
6. *Служцкий Е.* Элементы математической логики и теория множеств / *Е. Служцкий, Л. Борковский*. – М. : Прогресс, 1965.

Поступила 31.01.2011р.