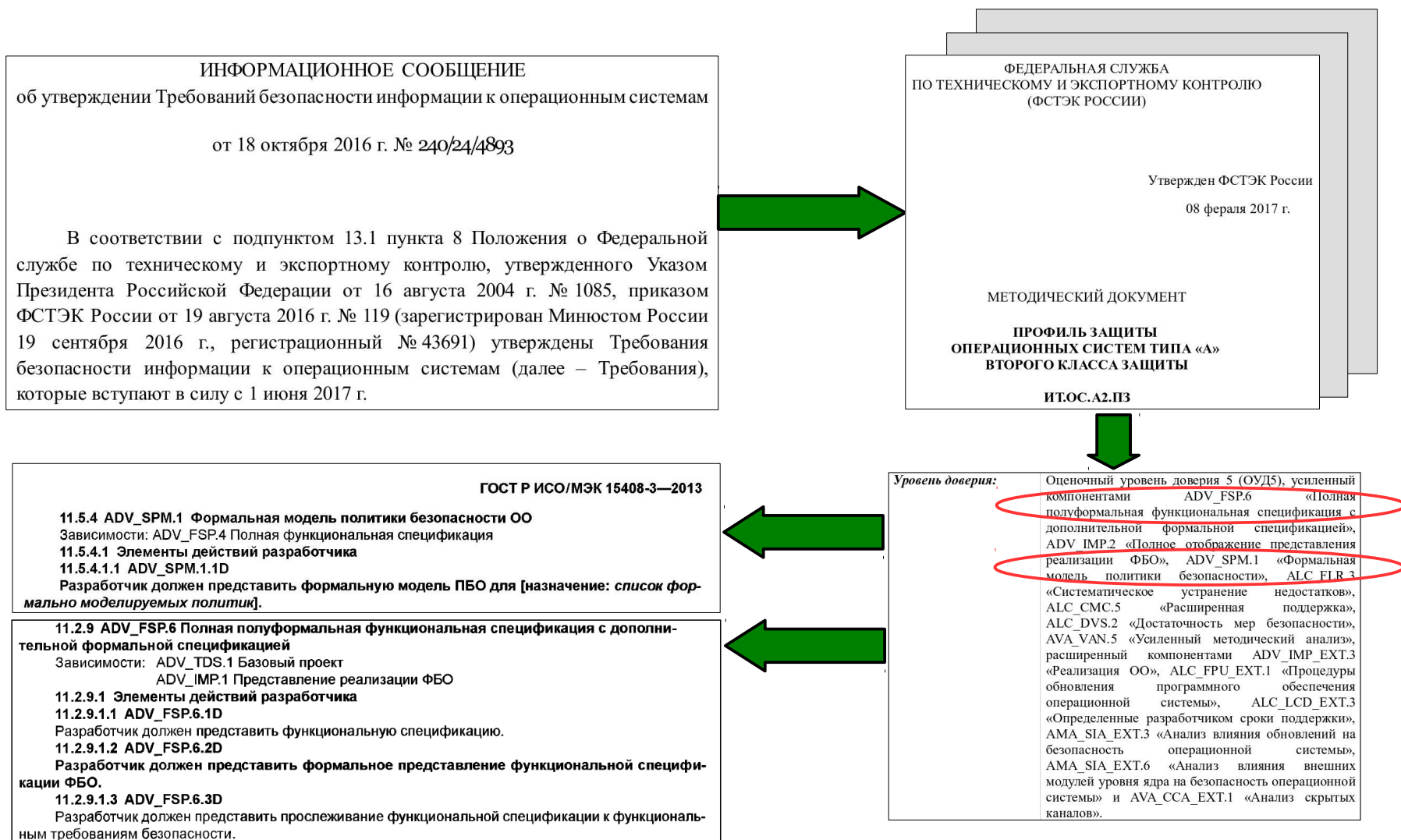


Научно-технологическая платформа моделирования и верификации политик безопасности управления доступом в операционных системах

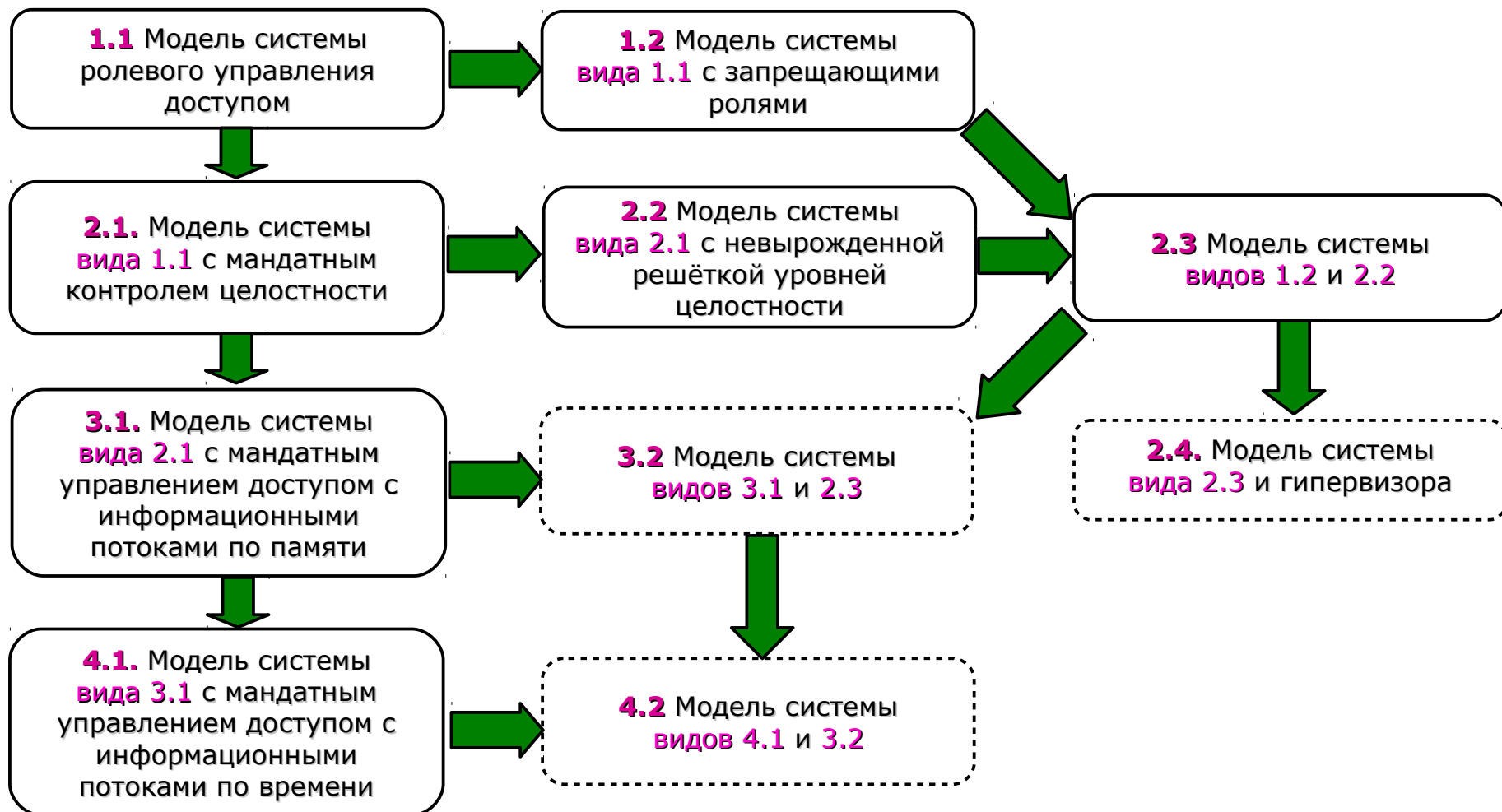


Девянин П.Н., д.т.н., доцент, ФУМО ВО ИБ
Оружейников А.Л., ООО «РусБИТех-Астра»
Хорошилов А.В., к.ф.-м.н., ИСП РАН

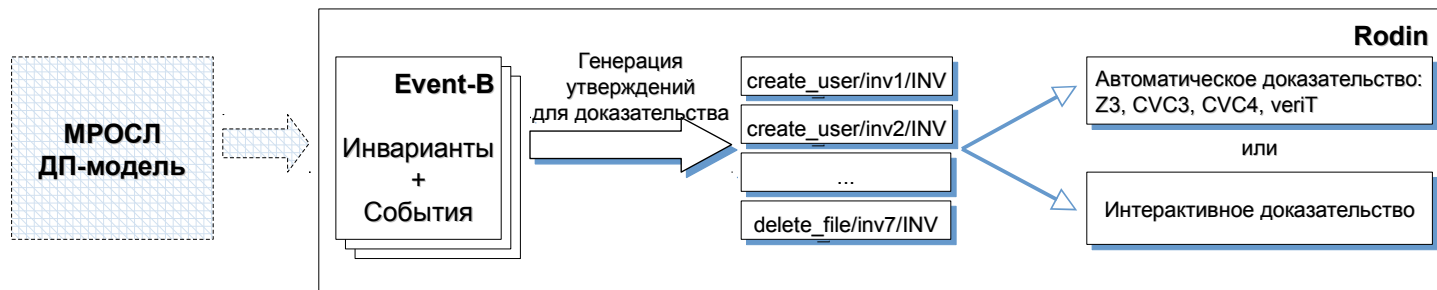
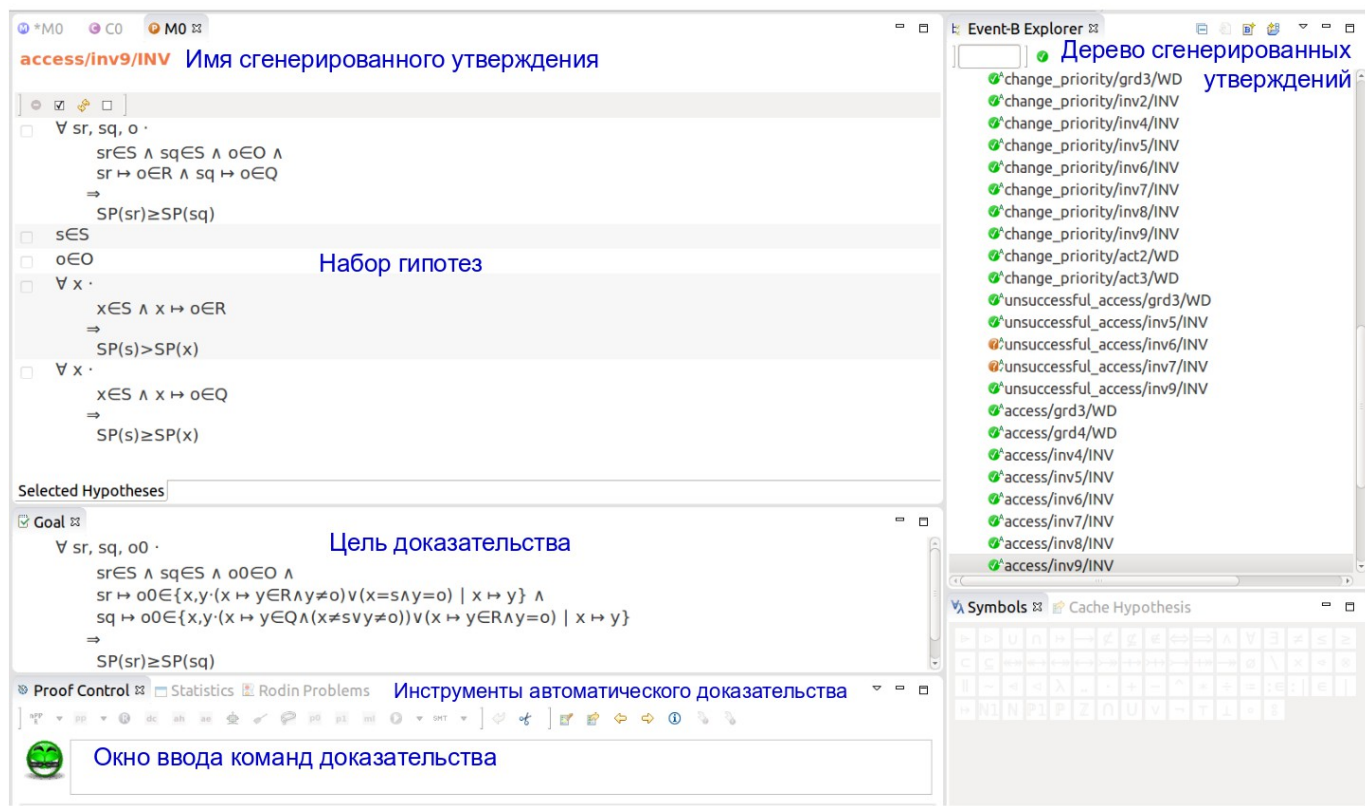
Нормативные документы ФСТЭК России 2017 г.



1. Иерархическое представление МРОСЛ ДП-модели в математической нотации



2. Дедуктивная верификация МРОСЛ ДП-модели в формализованной нотации Event-B (Rodin)



3. Дедуктивная верификация реализации МРОСЛ ДП-модели на языке С в среде Why3

Context

☐ Unproved goals
 ☒ All goals

Strategies

Compute

Inline

Split

Provers

Alt-Ergo (1.01 -Em)

Alt-Ergo (1.01 -trigs)

Alt-Ergo (1.30)

CVC4 (1.4)

CVC4 (1.4 noBV)

CVC4 (>=1.4 '14)

Eprover (1.9.1-001)

Spass (3.9)

Theories/Goals

	Status	Time
lab17_moore_majority.mlw	?	
Axiomatic_CountAxiomatic	?	
> Lemma CountOneHitAlt	✓	0.01
> Lemma CountOneMissAlt	✓	0.01
> Lemma CountNonNegative	?	
> Lemma CountMonotonic	?	
> Lemma CountUpperBound	?	
Function_moore_majority_safety	✓	0.63
Function moore_majority, safety	✓	0.63
Alt-Ergo (1.01 -Em)	✓	0.17
> split_goal_wp	✓	0.46
Function_moore_majority_behaviors	✓	3.81
Function moore_majority, default beh	✓	3.56
> split_goal_wp	✓	3.56
> 1. loop invariant init	✓	0.03
> 2. Assertion in line 56	✓	0.01
> 3. Assertion in line 57	✓	0.01
> 4. loop invariant preservation	✓	2.75
> 5. loop invariant preservation	✓	0.50
> 6. loop invariant preservation	✓	0.23
> 7. Ensures clause (Postcondition)	✓	0.03
Function moore_majority, behavior ele	✓	0.25

Source code

Task

Edited proof

Prover Output

Counter-example

file: /home/mordax/Downloads/lab17_moore_majority.c

```

32 /*@ requires size > 0;
33    requires valid(a + (0..size-1));
34    assigns \nothing;
35    ensures \exists integer i; 0 <= i < size && a[i] == \result;
36    behavior element_exists:
37      assumes \exists int i; count(a, size, i) > size / 2;
38      ensures count(a, size, \result) > size / 2;
39 */
40 int moore_majority(int *a, unsigned size)
41 {
42   int el = a[0];
43   unsigned cnt = 0;
44   /*@ loop invariant 0 <= i <= size;
45      loop invariant 0 <= cnt <= i;
46      loop invariant \exists integer j; 0 <= j < size && el == a[j];
47      loop invariant count(a, i, el) >= cnt;
48      loop invariant count(a, i, el) - cnt <= (i - cnt) / 2;
49      loop invariant \forall int x; x != el ==> count(a, i, x) <= (i
50      loop variant size - i;
51 */
52   for (unsigned i = 0; i < size; i++) {
53     if (cnt == 0) {
54       el = a[i];
55       cnt = 1;
56       /*@ assert count(a, i, el) >= 0;
57          /*@ assert count(a, i + 1, el) >= 1;
58     } else if (el == a[i]) {
59       cnt++;
60     } else cnt--;
61   }

```

Солверы

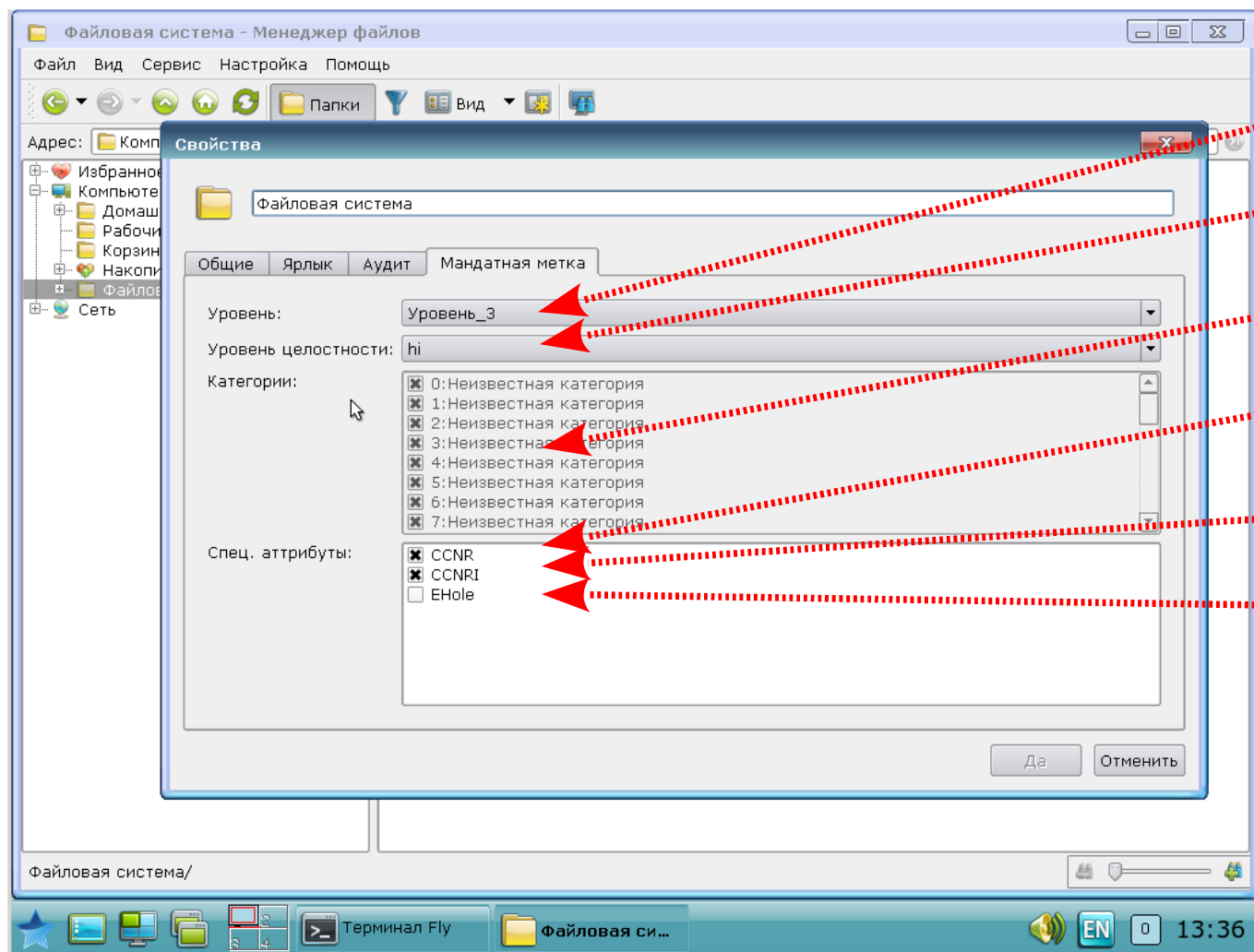
Условия верификации

Исходный код со спецификациями

Свойства
безопасности

Функциональные
свойства

4. Реализация MPROSL ДП-модели в коде ОСCH Astra Linux Special Edition



уровень
конфиденциальности

уровень целостности

неиерархические
категории

мандатный атрибут
конфиденциальности

мандатный атрибут
целостности

сущность-«дырка»

Научно-технологическая платформа



ИСПРАН

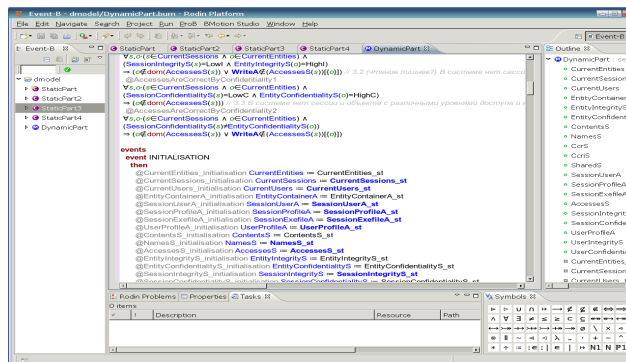
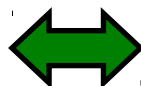
**Автоматизированная дедуктивная верификация
МРОСЛ ДП-модели в формализованной нотации**

Математическая нотация МРОСЛ ДП-модели

access_read(x, x', y)

$x, x' \in S, y \in E \cup R \cup AR$, существует $r \in R \cup AR$: $(x, r, read_a) \in AA$,
[если $y \in E$, то $(y, read_e) \in PA(r)$ и либо $(execute_container(x, y) =$
 $true$ и $f_e(y) \leq f_s(x)$], либо $(x, downgrade_admin_role, read_a) \in AA$,
[если $y \in R \cup AR$, то $(y, read_e) \in APA(r)$, $i_e(y) \leq i_s(x)$,
 $Constraint_{AA}(AA') = true$, (для $e \in y$] либо $(x, e, read_a) \in A$, либо $(x,$
 $e, write_a) \in A$, (либо $f_e(y) \leq f_s(x)$, либо $(x, downgrade_admin_role,$
 $read_a) \in AA$],
[если $y \in R \cup AR$ и $i_e(y) = i_high$, то $(x', f_s(x))_i_entity, write_a) \in A$

$S' = S, E' = E, APA' = APA, PA' = PA,$
 $user' = user, H_e' = H_e, F' = F,$
если $y \in E$, то $[A' = A \cup \{(x, y, read_a)\},$
 $AA' = AA,$
если $y \in R \cup AR$, то
 $[AA' = AA \cup \{(x, y, read_a)\}, A' = A]$



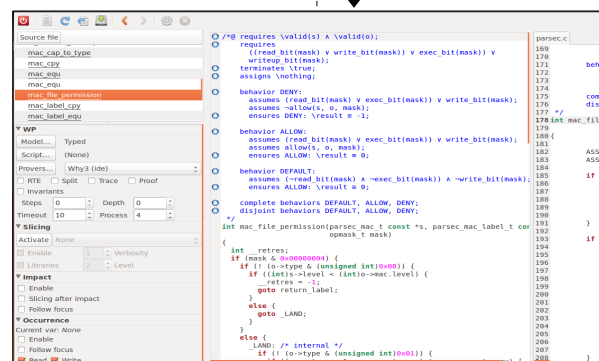
Rodin (Event-B)



**Требования
ADV_SPM.1, ADV_FSP.6**



```
1 static int access_read(struct task_struct *subject, struct inode *entity)
2 {
3     struct task_security *subject_security;
4     struct inode_security *entity_security;
5     int ret = -EACCES;
6
7     subject_security = get_task_security(subject);
8     entity_security = get_entity_security(entity);
9
10    if(!subject_security || !entity_security)
11        return ret;
12
13    if(is_role(entity)) //проверяем возможность получения доступа на чтение к сущности
14    {
15        ret = can_access(&subject_security->roles, &entity_security->list, MAY_READ, 0);
16        if(ret != 0)
17            ret = can_access(&subject_security->admin_roles, &entity_security->list, MAY_WRITE, 0);
18        if(ret == 0)
19        {
20            ret = execute_container(subject, entity);
21            if(ret != 0)
22                ret = is_downgrade_admin_role(&subject_security->admin_roles, MAY_READ);
23        }
24    }
25    else //проверяем возможность получения доступа на чтение к роли или административной роли
26    {
27        ret = can_admin_access(&subject_security->admin_roles, &entity_security->list, MAY_READ);
28    }
29    return ret;
30 }
```



Frama-C (Why3)

**Программный код механизма управления
доступом ОССН**

Верификации кода на соответствие спецификациям

Проект монографии «Базовый уровень МРОСЛ ДП-модели и его дедуктивная верификация»

1. Введение.
2. Анализ требований профилей защиты к представлению формальной модели политики безопасности.
3. Формирование научно-технологической платформы моделирования и верификации политик безопасности управления доступом.
4. Представление базового уровня МРОСЛ ДП-модели в математической нотации.
5. Дедуктивная верификация базового уровня МРОСЛ ДП-модели в формализованной нотации Event-B.
6. Подход к представлению формальной функциональной спецификации элементов механизма управления доступом ОССН.
7. Пример дедуктивной верификации в среде Why3 реализации на языке C элементов базового уровня МРОСЛ ДП-модели.
8. Заключение.

Спасибо за внимание!