

Д. Ю. Федоров, старший преподаватель кафедры «Вычислительных систем и программирования» СПбГЭУ

Т. М. Воробьев, студент направления «Прикладная математика и информатика» СПбГЭУ

ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ ИНФОРМАЦИОННО-ПСИХОЛОГИЧЕСКОГО ВОЗДЕЙСТВИЯ

«...важно понимать, какой силой обладают те, кто контролирует процесс выработки определений. Поэтому первый шаг в направлении установления контроля над определениями заключается в том, чтобы попытаться не уступить крайне важной терминологической территории»

(Г. Шиллер)

Доктрина информационной безопасности (далее – Доктрина) подчеркивает актуальность проблемы информационно-психологического воздействия на личность и общество. «В соответствии с военной политикой Российской Федерации основными направлениями обеспечения информационной безопасности в области обороны страны являются: ... нейтрализация информационно-психологического воздействия, в том числе направленного на подрыв исторических основ и патриотических традиций, связанных с защитой Отечества» [1].

Анализ ряда публикаций [2, 3, 4] показал, что оценка степени информационно-психологического воздействия на сегодняшний день носит, скорее, субъективный характер. В связи с этим возникает противоречие между необходимостью реализации Доктрины и отсутствием формализованной теории информационно-психологического воздействия.

Разрешением данного противоречия, на взгляд авторов, может стать предложенная далее теория информационно-психологического воздействия, основанием которой послужили труды проф. В. Я. Розенберга о семантических сетях знаний (далее – сетях знаний) [5, 6, 7].

Основные положения теории семантических сетей знаний профессора В. Я. Розенберга

Введем ряд определений. Прежде всего, под *мышлением* будем подразумевать отображение в мозгу человека общих существенных свойств (признаков) вещей, явлений внешнего мира (предметов мысли). Существенным признаком предмета называется тот признак, который выражает коренное, наиболее важное свойство предмета; если

существенный признак отсутствует, то предмет перестает быть данным предметом.

Под *знанием* в широком смысле будем понимать субъективный образ реальности в форме понятий и представлений [8].

Понятие – это мысль, которая отображает общие и существенные признаки предметов. Понятие отражает сущность вещи, имеет характер всеобщности. Одними и теми же понятиями пользуются разные люди. Понятие возникает и существует на базе языковых терминов и фраз. Определение понятия есть такое логическое действие, в процессе которого раскрывается содержание понятия. Раскрыть содержание понятия — это значит указать его существенные признаки. Каждый предмет имеет бесконечное число признаков, и пытаться указать все признаки предмета невозможно. Определение содержит в себе лишь такие признаки, которые, являясь существенными, отграничивают понятие от других понятий.

Под *представлением* будем понимать наглядный образ предмета. Представление всегда имеет индивидуальный характер, оно может складываться из несущественных признаков [9].

«...обучение происходит путем добавления новых концепций и предложений в существующую систему понятий, которой обладает обучаемый. Иногда возникает вопрос о происхождении первичных (базовых) понятий. Они приобретаются детьми в возрасте от рождения до трех лет, когда они распознают закономерности в окружающем их мире и начинают распознавать языковые метки или символы для этих закономерностей. Эта способность является частью эволюционного наследия людей. После 3-х лет новые знания получают путем вопросов и выяснения отношений между старыми и новыми понятиями» [10].

(Дэвид Пол Аусубель)

Совокупность знаний отдельного человека или всего человечества образует систему знаний. В качестве элемента системы знаний проф. В. Я. Розенберг [5] предложил использовать формализм понятия и технологию построения *сети знаний*. Основная идея сети знаний заключается в построении многоуровневой сети связанных между собой понятий. Понятия связываются через определения. Вышележащие понятия можно усвоить, если усвоены понятия, лежащие на более низком уровне сети знаний. В качестве примера на рис. 1 представлен фрагмент сети знаний, составленный на основе понятий из области уголовного права.

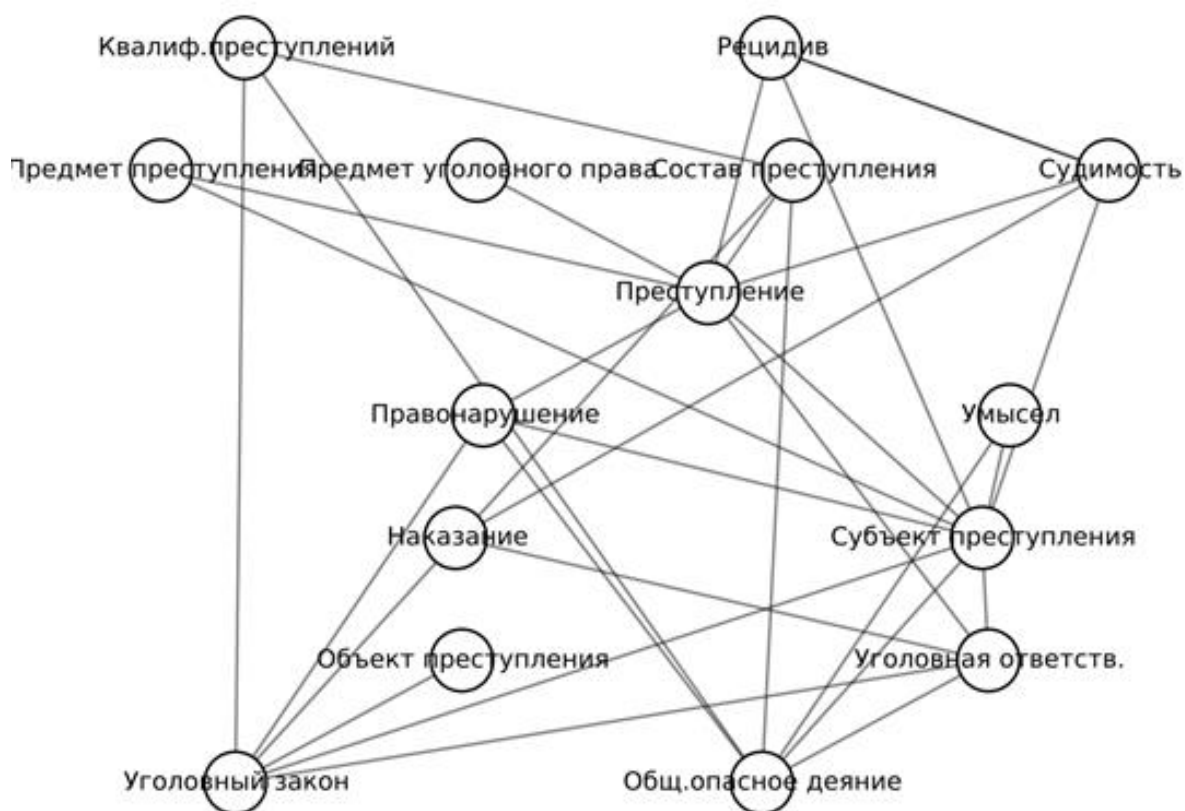


Рисунок 1. Фрагмент сети знаний для области уголовного права

Пример построения индивидуальной картины мира человека на основе сетей знаний

Человека всюду окружают информационные потоки, поэтому необходимо различать понятия «информация», как входной поток, и «знание», как переработанную информацию. Федеральный закон «Об информации, информационных технологиях и о защите информации» определяет информацию, как сведения (сообщения, данные) независимо от формы их представления. В терминах теории сетей знаний, под *знанием* будем понимать набор понятий и связей между ними. Исходя из этого определения, информация становится знанием в момент построения сети знаний.

Рассмотрим область индивидуального человеческого знания $K \subset U$, где K – множество понятий и их определений из множества U – всех знаний человечества на данный момент времени [6]. Условимся, что K является упрощенной моделью индивидуальной картины мира человека.

Для автоматизации построения индивидуальной картины мира на базе семантических сетей на языке Python [11] была написана программа

(автоматизированная система построения сети знаний)¹, реализующая следующий алгоритм извлечения знаний (см. рис. 2) [12].

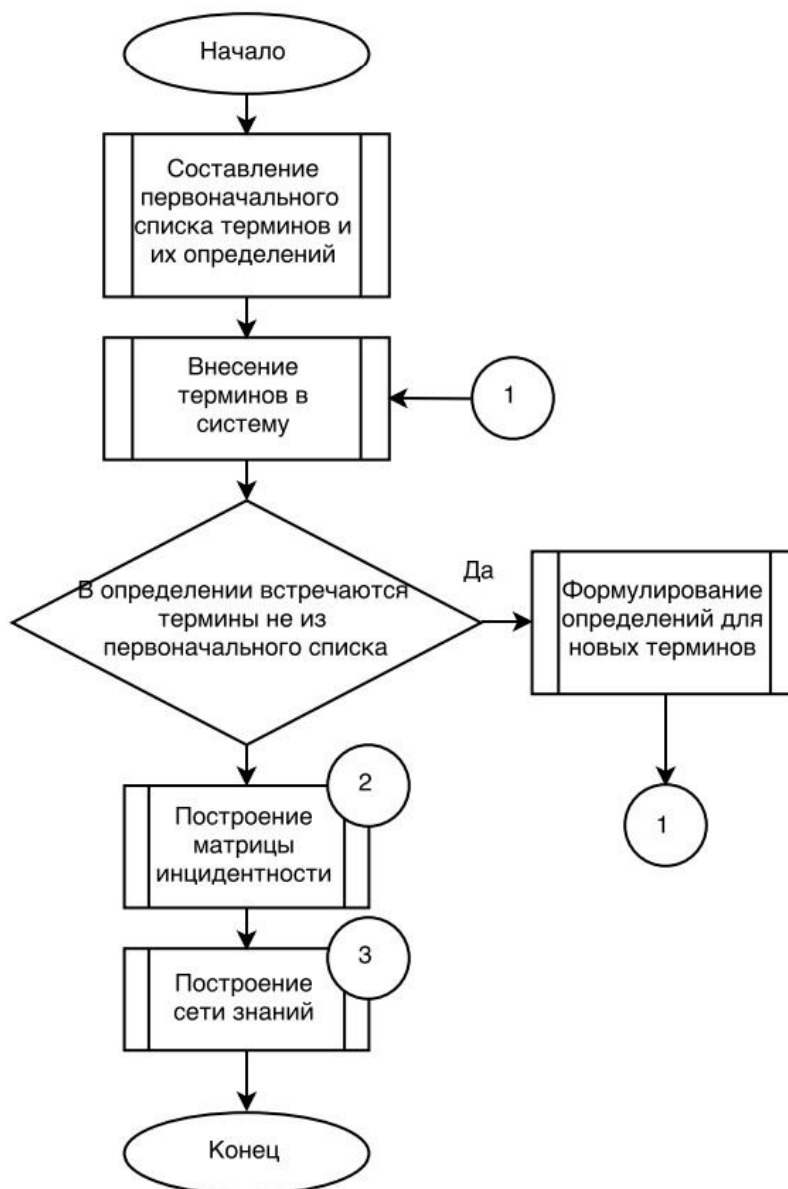


Рисунок 2. Блок схема алгоритма извлечения знаний для построения индивидуальной картины мира человека (сети знаний)

Разработанная автоматизированная система построения сети знаний решает следующие задачи:

- 1) позволяет с клавиатуры вносить определения терминов в систему;
- 2) позволяет добавлять в систему новые термины и их определения;
- 3) по терминам и их определениям строит матрицу инцидентности однонаправленного графа (сеть знаний);

¹ Программа реализована студентом направления «Прикладная математика и информатика» СПбГЭУ Т.М. Воробьевым.

4) по матрице инцидентности строит сеть знаний.

Для реализации обозначенных задач используются следующие библиотеки на языке Python: string для работы со строками, copy позволяет копировать объекты, numpy для работы с большими массивами, matplotlib позволяет визуализировать ориентированный граф (сеть знаний), networkx для создания и обработки графов, pymongo2 и nltk для обработки текстов, pymongo для работы с базой данных.

В основе работы автоматизированной системы построения сети знаний лежит клиент-серверная архитектура. Добавление терминов в систему производится через веб-интерфейс.

Перейдем к рассмотрению реализации основных элементов системы.

```
# составляем список из терминов, для которых потребуется
# сформулировать определения:
s1 = ['дерево свойств', 'интегративные свойства', 'качество', 'объект',
'поведение', 'подсистема', 'развитие', 'свойство', 'связь', 'система', 'ситуация',
'состояние', 'среда', 'структура', 'управление', 'функционирование',
'характеристика', 'целое', 'цель', 'элемент']
```

Исходный код обращения к базе данных для считывания терминов и их определений, введенных пользователем:

```
client = MongoClient('localhost', 27017)
db = client['main']
collection = db['laba']
s1 = []
s2 = []
nul = []
# содержит список терминов:
s1=list(collection.find_one()['dicts'][1].keys())
# содержит список определений:
s2=list(collection.find_one()['dicts'][1].values())
```

Удалим из всех терминов и их определений знаки пунктуации и не несущие нагрузки слова при помощи методов stopwords.words(«russian») и string.punctuation. Далее преобразуем все определения в список слов, входящих в них, и переведем все слова в нормальную форму. Это позволяет сделать функция tokenize:

```
# на вход функции подается строка, содержащая термин
# или его определение
def tokenize(s):
    # разбивает строку, написанную на русском языке,
    # на список из слов и знаков пунктуации, входящих в эту строку:
    tokens = nltk.word_tokenize(s,language='russian')
```

```

prepositions = stopwords.words("russian")
# создает новый список из слов, являющихся
# нормальными формами слов, входящих в строку,
# не входящих в список prepositions и не
# являющимися пунктуационными знаками:
tokens = [morph.parse(i)[0].normal_form for i in tokens if ( i not in
prepositions and i not in string.punctuation)]
return tokens

```

Вызов функции tokenize производится следующим образом:

```

for i in range(len(s1)):
    s1[i]=tokenize(s1[i])
    s1[i]=' '.join(s1[i])
    # превращает термин в строку, состоящую из слов
    # в нормальной форме:
    s2[i]=tokenize(s2[i])

```

Матрицей инцидентности однонаправленного графа называется матрица, в которой на позиции $[i][j]$ стоит 1, если элемент i связан с элементом j или 0 – в противном случае. В данном случае на позиции $[i][j]$ стоит 1, если определение термина j содержит термин i или 0 – в противном случае. Реализация построения матрицы (изначально матрица нулевая) имеет следующий вид:

```

nul = [0 for i in range(len(s1))]
matrix = []
labels = {}
for i in range(len(s1)):
    nule = copy.copy(nul)
    # создание нулевой матрицы размера len(s1) на len(s1):
    matrix.append(nule)
    labels[i] = s1[i]
    s1[i] = tokenize(s1[i])
    s1[i] = ' '.join(s1[i])
    s2[i] = tokenize(s2[i])
for i in range(len(s1)):
    for j in range(len(s2)):
        if (i != j):
            if len(s1[i].split()) == 1:
                for k in range(len(s2[j])):
                    if s1[i] in s2[j][k]:
                        matrix[i][j] = 1
                    elif stemmer.stem(s1[i]) in s2[j][k]:
                        c = input('Относится ли слово '+s2[j][k]+' к термину '+
labels[i]+'?
Да-1
Нет-2')
                        if c == '1':

```

```
matrix[i][j] = 1
else:
    for k in range(len(s2[j])-len(s1[i].split()+1):
        s3 = "
        for n in range(len(s1[i].split())):
            s3 = s3 + ' ' + s2[j][k+n]
        if s1[i] in s3:
            matrix[i][j] = 1
            break
```

Заметим, что отдельно рассматривается случай, когда термин составной. При данном решении возникают проблемы со сложными словами, например, со словом «целенаправленный». Также проблемы возникают со словами, которые образованы от терминов, например, «целостный» или «связанный», так как метод `normal_form` возвращает слово в нормальной форме, то есть слово «связанное» преобразуется в слово «связанный» или «связать», но не в слово «связь». Для решения этой проблемы пришлось воспользоваться методом `stemmer.stem(s)`, который возвращает основу слова, например, «целый» преобразуется в слово «цел». Но возникает новая проблема, так как термины, например, «целый» и «цель» имеют одинаковые основы. Для решения этой проблемы пришлось запрашивать у пользователя, к какому термину относится, например, слово «целостный» – «цель» или «целый».

Далее представлен пример обработки терминов пользователем:

Относится ли слово связанный к термину связь

Да

Нет

Пользователь - Да

Относится ли слово целостный к термину целое

Да

Нет

Пользователь - Да

Относится ли слово целенаправленный к термину целое

Да

Нет

Пользователь - Нет

Относится ли слово посредством к термину среда

Да

Нет

Пользователь - Нет

Относится ли слово целостный к термину цель

Да

Нет

Пользователь - Нет

Относится ли слово целенаправленный к термину цель

Да
Нет
Пользователь - Да

В начале процесса построения сети знаний создается ориентированный граф G и в него добавляется количество вершин равное длине матрицы. Хранить информацию о каждой вершине будем в списке `graf`, где:
`graf[i][0][0]` – уровень i -ой вершины (изначально равен '0'),
`graf[i][1]` – список вершин, связанных с вершиной i ,
`graf[i][2]` – список вершин, с которыми связана вершина i .

```
G=nx.DiGraph()
G.add_nodes_from([i for i in range(len(matrix))])
# словарь, который будет хранить координаты в виде pos[i]=(x, y)
pos={}
# переводит список matrix в массив:
matrix=numpy.array(matrix)
graf=[]
for i in range(len(matrix)):
    graf.append(['0',[],[]])
# максимальный уровень графа
maximum=0
```

Заметим, что в начале программы был создан словарь `labels`, который хранит не отформатированные термины – названия вершин графа:

```
labels = {}
for i in range(len(s1)):
    labels[i] = s1[i]
```

Необходимо найти все корни, то есть элементы, с которыми никакие другие элементы не связаны. Из определения матрицы инцидентности однонаправленного графа следует, что элемент j является корнем, если j -ый столбец матрицы инцидентности нулевой. Найдем все такие элементы и присвоим им нулевой уровень, а список входящих вершин изменим на ['Нет']:

```
j=0
for i in matrix.transpose().tolist():
    if 1 not in i:
        graf[j][0][0]=0
        graf[j][1].append('Нет')
        pos[j]=(j,0)
        j=j+1
```


Далее, опираясь на правила построения сети знаний, формируем ориентированный граф:

- а) каждая вершина должна находиться на определенном уровне;
- б) если вершина i связана с вершиной j , то вершина j должна, по крайней мере, находиться на один уровень выше уровня вершины i ;
- в) если вершина i не связана ни с какой из вершин j и не находится на последнем уровне, то ее необходимо переместить на последний уровень.

Граф строится при помощи обхода в ширину. Счетчик k обозначает, какой уровень рассматривается. Если элемент j лежит на уровне k , то рассматриваем все i -ые вершины, с которыми связана вершина j . Затем в граф G добавляем ориентированное ребро – $G.add_edge(j, i)$, где j – начало, а i – конец и обновляем списки выходящих вершин для вершины j и список входящих вершин для вершины i . Если уровень вершины i больше уровня вершины j , то ничего не делаем. В противном случае, так как граф должен соответствовать пункту б, то применяем к i вершине рекурсивную функцию up , которая поднимает все поддерево с корнем в вершине i на величину равную разности уровней вершин j и i плюс один:

```
def up(i,q):
    global graf,maximum,pos
    graf[i][0][0] = q+1
    pos[i] = (i,graf[i][0][0])
    if graf[i][2] != []:
        for j in graf[i][2]:
            up(j,graf[i][0][0])
    elif graf[i][0][0]>maximum:
        maximum = graf[i][0][0]
j=0
for i in matrix.transpose().tolist():
    if 1 not in i:
        graf[j][0][0]=0
        graf[j][1].append('Нет')
        pos[j]=(j,0)
    j=j+1
k=0
while k<=maximum:
    for j in range(len(matrix)):
        if graf[j][0][0]==k:
            for i in range(len(matrix)):
                if matrix[j][i]==1:
                    G.add_edge(j,i)
                    graf[j][2].append(i)
                    graf[i][1].append(j)
```

```

        if int(graf[i][0][0])<=graf[j][0][0]:
            up(i,graf[j][0][0])
    k=k+1
for i in range(len(graf)):
    if graf[i][2]==[]:
        graf[i][0][0]=maximum
        pos[i]=(i,maximum)

```

Также рассматриваем уровень верхних элементов поддерева в корне i , после их поднятия, и сравниваем его со значением `maximum`, которое отвечает за значение максимального уровня графа. Последний цикл, удовлетворяет требованию v . Он поднимает все вершины, не имеющие детей, на максимальный уровень.

Улучшение визуализации ориентированного графа:

```

for i in range(maximum+1):
    l=[]
    for j in range(len(graf)):
        if pos[j][1]==i:
            l.append(pos[j])
    j=0
    while j < len(l)-1:
        if l[j+1][0]-l[j][0]<3:
            pos[l[j+1][0]]=(pos[l[j][0]][0]+3,i)
            l[j+1]=(l[j][0]+3,i)
            l.sort()
        else:
            j=j+1

```

Визуализации ориентированного графа:

```

nx.draw_networkx(G,pos,with_labels=False,node_size=2000,node_color='w',
node_shape='s')
nx.draw_networkx_labels(G,pos,labels,font_size=8)
plt.axis('off')
plt.show()

```

Результат выполнения программы представлен на рис. 3, 4 и 5.

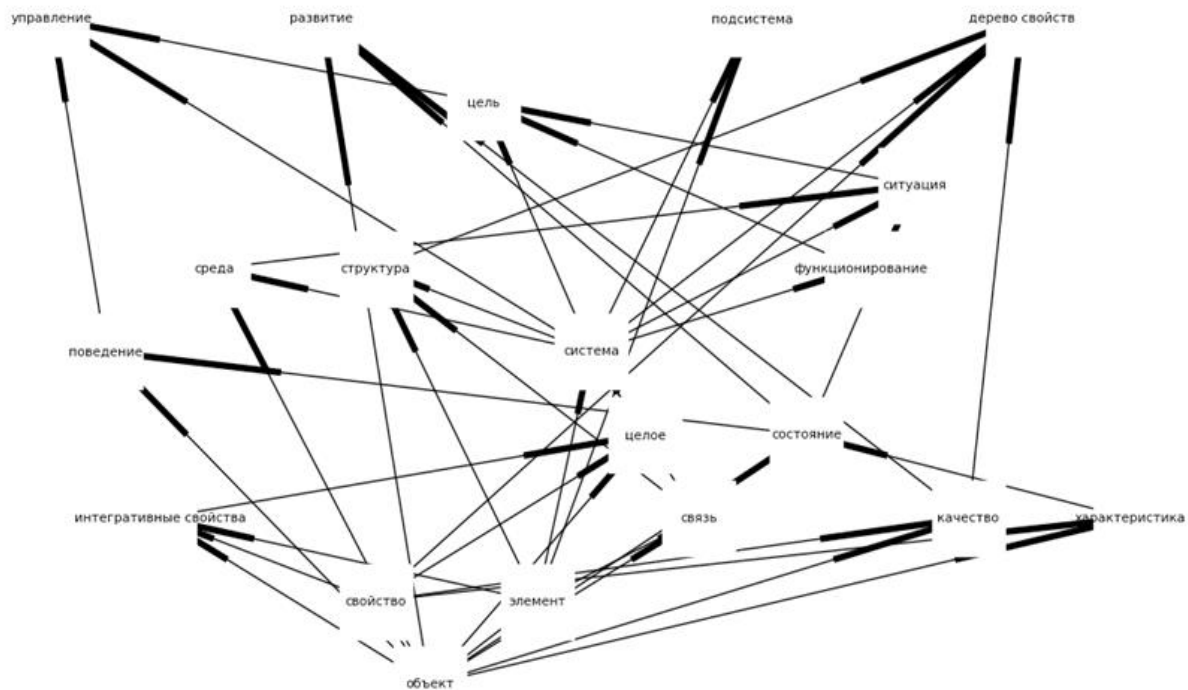


Рисунок 3. Пример построения сети знаний для терминов из предметной области «Теория систем»

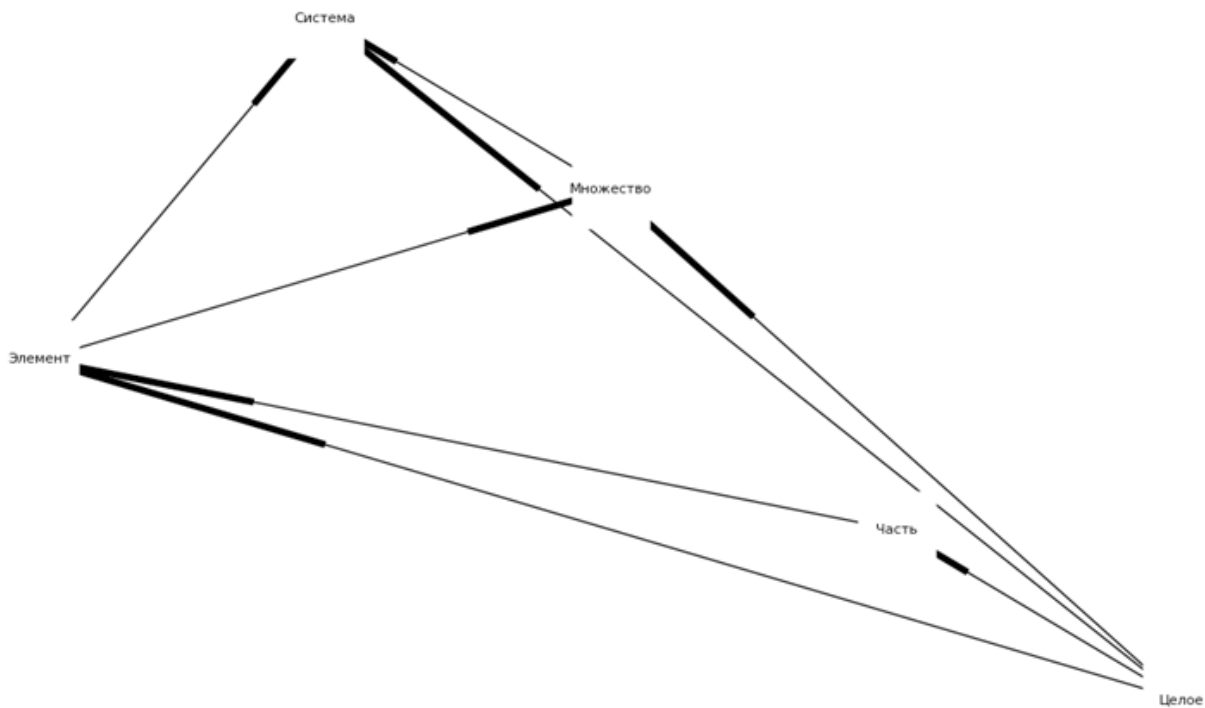


Рисунок 4. Пример построения сети знаний

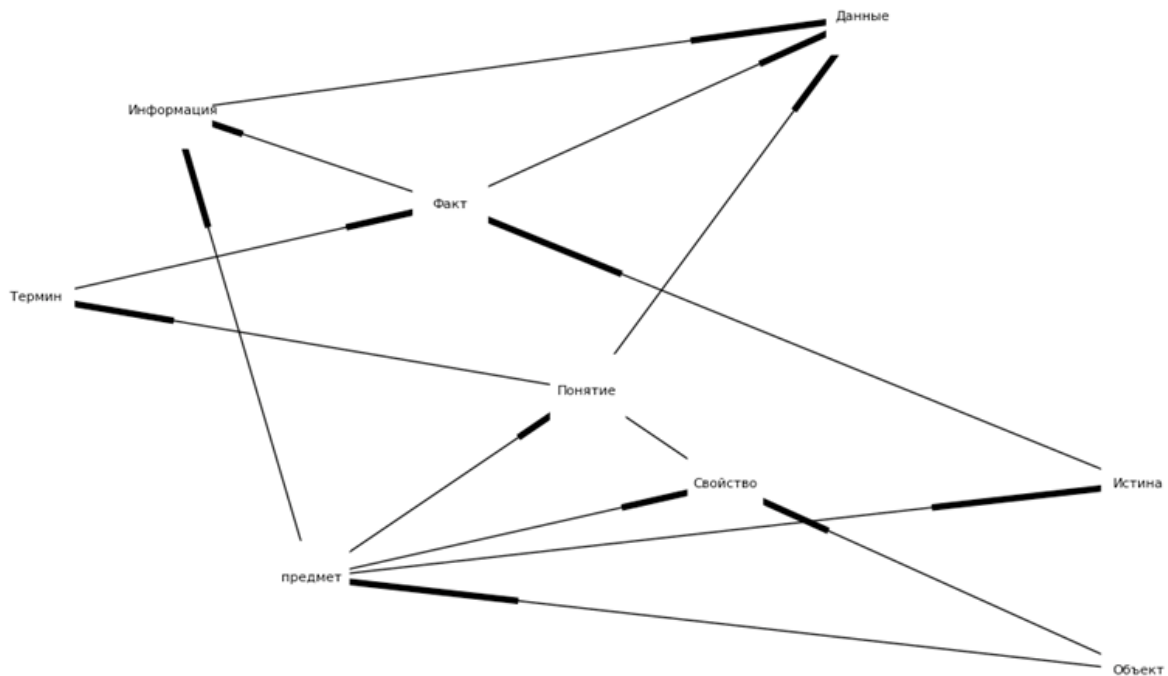


Рисунок 5. Пример построения сети знаний

Основные положения теории информационно-психологического воздействия

Субъект (человек, группа людей) по средствам информации T (в частности, через текст) производит воздействие на объект (человека, группу людей), который данную информацию воспринимает посредством органов чувств. Информация формируется таким образом, чтобы максимально воздействовать (деформировать, исказить) на модель знаний K (индивидуальную картину мира), содержащуюся в голове объекта воздействия. Исходя из этого, можно сформулировать понятие *количества энтропии информации* (текста), как величину, определяющую степень деформации модели знаний объекта (*z-энтропия*).

Рассмотрим способы осуществления информационно-психологического воздействия на примере сетей знаний, в основу которых положена классификация Леонтьева А. А. [13].

1. «Ввести в поле значений (в контексте сети знаний, картину мира K) реципиента новые значения, сообщить ему такие новые знания о действительности, на основе которых он изменит свое поведение или, по крайней мере, свое отношение к этой действительности». Примером такого воздействия может быть новость о взрыве во время проведения Бостонского марафона 15 апреля 2013 года. Это событие актуализировало понятие, которое в картине мира реципиента ассоциировалось со взрывом. Например, в зависимости от индивидуальных представлений ассоциации

могли быть следующими: «теракт для устрашения», «злая шутка», «заговор спецслужб», «месть воинов джихада» и др. В результате, актуализированное понятие может стать мишенью для дальнейшей манипуляции.

2. «...изменить поле значений реципиента, не вводя в него новых элементов, т.е. изменить понимание реципиентом событий и их взаимосвязи. Это тоже информирование, но на другом уровне, когда событие уже известно, но благодаря воздействию оно интерпретируется реципиентом по-другому». Примером такого воздействия может быть сообщение об организаторах Бостонского теракта. Со слов председателя Комитета Палаты представителей по национальной безопасности: «Поездка Тамерлана Царнаева в северо-кавказский регион, радикальные видео о провозглашении халифата, которые он разместил в Интернете по возвращении, взрывные устройства, которые были использованы им и его младшим братом, – все это говорит о том, что этот теракт был инспирирован «Аль-Кайдой»». Реципиенту известен факт совершения взрыва (см. п.1), ранее было известно о существовании «Аль-Кайды». Через информационное сообщение устанавливается связь между двумя понятиями: актуализированным после новости о событии (см. п.1) и ранее известным – «Аль-Кайда». Например, в зависимости от актуализированного понятия в п.1, «Аль-Кайда» в индивидуальной картине мира реципиента может связываться с «террористами, действующими для устрашения», «злой шуткой», «заговором спецслужб», «воинами джихада» и др.

3. В картину мира реципиента вводится новый элемент, который отсутствовал там ранее. Например, сообщение о братьях Царнаевых, подозреваемых в организации взрывов на Бостонском марафоне. Впоследствии с новым элементом устанавливается связь, аналогичная п.2. Таким образом, картина мира реципиента достраивается новыми элементами.

4. В картине мира реципиента старое понятие подменяется новым, т.е. известное понятие обогащается информацией, ранее о которой не было известно реципиенту. Например, Плутон, когда-то был девятой планетой, а сегодня является представителем нового семейства планет-карликов. Изменилось содержание понятия «планета».

5. В картине мира реципиента старое понятие вытесняется новым, не схожим по смыслу со старым. Например, казнокрадство именуется «нецелевым расходованием бюджетных средств». Замещение понятия приводит к перестроению ассоциативного ряда реципиента. Новое понятие не несет негативной окраски, присущей старому понятию.

На основании алгоритма построения сети знаний можно ввести определение ранжирования понятия. Под *ранжированием* будем понимать

определение порядка понятий согласно их авторитету. Авторитет понятия формируется по количеству вхождений в определения других понятий из области знаний K . Наиболее авторитетные понятия составляют множество $A \in K$ и будут являться системообразующими для области знаний K [13].

Любое воздействие на множество K переводит его из состояния S_1 в состояние S_n , где S_n – набор актуализированных понятий на данный момент времени. Приведенная классификация видов информационного воздействия является неполной, но позволяет построить модель формирования индивидуальной картины мира K под воздействием поступающей информации T . Главной мишенью информационного воздействия является множество системообразующих понятий A , изменение которых приведет к перестроению множества K , т.к. системообразующие понятия включаются в определения большого числа других понятий.

Следует отметить, что рассмотренные положения теории информационно-психологического воздействия находятся на начальном этапе формирования².

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Доктрина информационной безопасности РФ (№ Пр-646 от 5 декабря 2016 г.).
2. Соловей В. Д. Абсолютное оружие. основы психологической войны и медиаманипулирования. М.: Издательство «Э», 2015. — 320 с. (fake. технология фальсификаций).
3. Володенков С. В. Интернет-коммуникации в глобальном пространстве современного политического управления. — Издательство Московского университета Москва, 2015. — С. 272
4. Макаренко С.И. Информационное противоборство и радиоэлектронная борьба в сетевых войнах начала XXI века. Монография. — СПб.: Научно-технологические исследования, 2017. — 546 с.
5. Розенберг В. Я. Аксиомы математической теории исчисления знаний: статья. Международный научный институт «Educatio». Ежемесячный научный журнал, № 5(12) / 2015, ч. 3.
6. Розенберг, В. Я. Система обучения на базе семантических сетей. Теория и практика // Фундаментальные и прикладные исследования в современном мире: материалы Междунар. науч.-практ. конф. СПб.: Информ. изд. учеб.-науч. центр «Стратегия будущего», 2013. С. 184–191.

² Список публикаций по теме исследования: <http://dfedorov.spb.ru/science.html>

7. Федоров, Д. Ю. Кибернетический подход к управлению процессом обучения на основе семантических сетей знаний. СПб.: Изд-во Политех. ун-та, 2016. 40 с.
8. Знание. Статья в Википедии. [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/Знание> (дата обращения 17.10.2017)
9. С. Н. Виноградов, А. Ф. Кузьмин. Логика (учебник для средней школы), М.:Учпедгиз, 1954.
10. Joseph D. Novak Learning, Creating, and Using Knowledge: Concept maps as facilitative tools in schools and corporations.
11. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – М. : Издательство Юрайт, 2017. – 126 с.
12. Подружкина, Т. А. Алгоритмы планирования процесса обучения на основе семантических сетей знаний / Т.А. Подружкина, Д.Ю. Федоров // Научно-аналитический журнал «Вестник Санкт-петербургского университета ГПС МЧС России». - 2017. - № 1. - С. 107-116
13. Федоров, Д. Ю. Применение структуризации знаний для обеспечения информационной безопасности личности // Национальная безопасность и стратегическое планирование. — 2013. — № 2. – С.23-27