

מחלקת הנדסת תוכנה

שם הפרויקט: סוכן רשת אוטומטי

Project Name: Automatic Web Agent

דו"ח ביניים 2

|  |  |
| --- | --- |
| שם הסטודנט: | קובי הרשקוביץ |
| מספר תעודת זהות: | 025632654 |
| שם המנחה: | אלעד הוגן |
| תאריך ההגשה: | 00/00/10 |

1. תוכן ענינים
2. תקציר

השימוש באינטרנט נעשה רווח כמעט בכל בית ובאמצעותו ניתן לבצע משימות רבות ומגוונות החל מקבלת מידע בידורי וכלה בפעולות בנקאיות. במרבית מן הפעולות נדרש המשתמש להזדהות, לבצע מעקב אחר נתונים באתר, ולבצע פעולות על סמך ידע שלמד מאתרים אחרים. פעולות אלו במרבית הזמן זהות/דומות בין האתרים וכמו כן ביצוען עלול לדרוש תשומת לב וזמן רב מן המשתמש.

הפרוייקט הינו סוכן רשת אוטומטי הבא להחליף מטלות שגרתיות וגוזלות זמן של משתמש דפדפן ממוצע/מתקדם. באמצעות המערכת המוצעת יוכל המשתמש לתכנן תסריט של פעולות מסויימות באתרים מסויימים בהתאם למידע הנדגם מאתרים אחרים ולקבוע את זמן ביצוע התסריט.

לדוגמא ניתן לקבוע תסריט שבו נדגם כל 4 שעות אתר סקי שהמשתמש מחבב במיוחד (לדוגמא www.champery.com) , התסריט מכוון לזהות את השדה של גובה השלד באתר וברגע שגובה השלג מגיע ל-110cm התסריט שולף מאתר ה-"דקה ה-90" את מחירי חבילות הנופש לאתר ושולח דוח מפורט עם הנתונים באי-מייל למשתמש.

דוגמא נוספת יכולה להיות תסריט שיאריך את זמן השאלת ספר באתר הספריה של מכללת אפקה ,שאינו מצוייד כיום בסוכן אוטומטי להארכת השאלות, פעם בשלושה ימים.

יחודיות הפרוייקט היא שבניגוד לסוכנים אוטומטיים הנמצאים באתר מסויים ונשלטים מתוך האתר הפרוייקט הנ"ל מציע סוכן שיתוכנת בצד הלקוח ולא בצד השרת, דבר אשר יאפשר לו נוחויות ושליטה מירבית להשגת מטרותיו.

מטרת דו"ח ביניים 2 הינה להציג את התקדמות פרוייקט הגמר – סוכן רשת אוטומטי (Automatic Web Agent). את תוצרי התיכון המפורט ניתן לחלק ל-4 קבוצות:

* **סעיפים המתייחסים לתיכון המערכת**
  + ניתוח חלופות לתיכון – תיאור השוואתי של מטודולוגיית התיכון שנבחרה אל מול אלטרנטיבות אפשריות.
  + הפתרון שנבחר – סעיף המציג את תמצית הפתרון שנבחר לתיכון המערכת, עם הפנייה למסמך SDD מפורט.
  + סקר ספרות – תיאור תמציתי של נושאים בהם ישנו דגש מבחינת השלמת פערי ידע, או מקורות התייחסות ספרותיים הנדרשים לצורך השלמת הפרוייקט.
  + תוכנית בדיקות – תמצית תוכנית הבדיקות אשר תהווה מדד איכותי לעמידת המערכת בדרישות, עם הפנייה למסמך STD מפורט.
* **סעיפים מערכתיים**
  + לוחות זמנים – תיאור לוח הזמנים המעודכן של הפרוייקט.
  + ניהול סיכונים מעודכן – הצגת תוצרי החשיבה של תהליך ניתוח הסיכונים של המערכת, ודרכי ההתמודדות האפשריות עם סיכונים אלו.
* **נספחים**
  + **מסמך SRD – Software Requirements Document**

מסמך המתאר את הדרישות הפונקציונליות והמערכתיות אשר בהן מערכת התוכנה צריכה לעמוד (מבוסס על IEEE standard 830).

* + **מסמך SDD – Software Design Description**

מסמך המתאר את תיכון המערכת וכולל את ארכיטקטורת התוכנה, רכיבי התוכנה השונים ופירוט ממשקיהם, ותרשימי UML ו-Use Cases.

(מבוסס על IEEE standard 1016).

* + **מסמך STD – Software Test Documentation**

תיאור תיכון תוכנית הבדיקות של המערכת, אשר תשמש כמדד איכותי לעמידה בדרישות המערכת. (מבוסס על IEEE standard 829).

* + **מסמך SPMP – Software Project Management Plan**

מסמך זה מתאר מרכיבים של ניהול סיכונים ופרמטרים נוספים הרלוונטים לניהול הפרוייקט. (מבוסס על IEEE standard 1058).

הפרוייקט נמצא, בעת כתיבת המסמך, בשלבי סיום של שלב התיכון ולקראת סוף מימוש אב הטיפוס, ויחל את שלב מימוש המערכת עם תום שלב זה.

1. מטרות עבודה

רבות מהפעולות הפיננסיות, הלימודיות והבידוריות נעשות באמצעות דפדפן ברשת האינטרנט:

• רישום לאתרים

• קניה/מכירה של מניות

• הזמנת כרטיסים

• הגשת מטלות

• בחירת קורסים

רבות ממטלות אלו הן מחזוריות, שגרתיות ועלולות להתיש את המשתמש ו/או לגרום לו לבצע שגיאות באופן הכנסת הנתונים.

מטרת הפרוייקט הנגזרת מבעיה זו היא בניית אפליקציה אשר תבצע מטלות שגרתיות וצפויות מראש בצורה אוטומטית במקום משתמש דפדפן אנושי

ניתן לפרוט מטרת על זו ל-3 יעדים:

• ניטור אתר בצורה מחזורית

• ביצוע פעולות אוטומטיות בדפדפן

• התראה למשתמש ע"פ בקשתו על אירוע באתר מסויים.

1. סקירה ספרותית

לא נוספו חומרים נוספים לסקירה מעבר למוזכר ב-SRD. [‏11.1 [

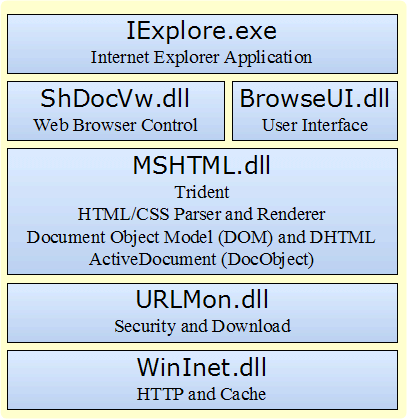
1. אלטרנטיבות לתיכון המערכת
   1. התממשקות לדפדפן:

כדי לבצע ניתור ושליטה בדפדפן נבחנו מספר דרכים להתממשקות לדפדפן בצורה תכנותית.

ישנם מספר ממשקי תוכנה (API) אשר מאפשרים גישה ושליטה בדפדפן ובמבני הנתונים אותם הוא מנהל:

* + 1. מודול mshtml.dll של IE

מודול זה, הנקרא לעיתים "Trident" [], נמצא בלב הדפדפן Internet Explorer כפי שניתן לראות מ, והוא אחראי לניהול הניתוח וההצגה של HTML ו-CSS. מודול זה חושף ממשקים אשר מאפשרים לשלוט ברכיבים שהדף מציג.



איור

מכיוון שמודול זה ממומש כ-COM object אזי ניתן להשתמש בו כ-reference בכל שפת NET.

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| ממשק מוכר ובדוק | ממשק שיושב "על הברזלים" ולכן יש מימוש מסובך עבור שליטה בכל אלמנט |
| מהיר | תומך אך ורק ב-IE |
|  |  |

* + 1. ספריית הקוד הפתוחWatiN

WatiN [http://sourceforge.net/projects/watin] היא ספריית קוד פתוח המאפשרת לבצע אוטומציה של דפדפנים באמצעות שפות NET.

מאפשרת הרצת בדיקות אוטומטית של אתרים באמצעות אינטראקציות עם הדפדפן. היא כתובה ב-C# אבל ניתן להשתמש בה בכל שפה ממשפחת NET.

רשימת התכונות הבולטות של ספרייה זו:

* + אוטומציה של כל הלמנטים המרכזיים ב-HTML.
  + חיפוש אלמנטים באמצעות תכונות מרובות.
  + תמיכה ב-AJAX
  + תמיכה ב-frames(cross domain) וב-iframes.
  + תמיכה בדיאלוגים מסוג popup כגון alert, confirm, login וכו'
  + תמיכה ב-HTML dialogs
  + מימוש מודל של Page, Control.
  + תמיכה ביצירת תצלומי מסך של דפים.
  + מנגנון logging.
  + שילוב פשוט עם כלי unit test.
  + תמיכה בדפדפנים: IE 6,7,8/Firefox 2,3
  + קוד פתוח ([Apache License 2.0](http://www.apache.org/licenses/LICENSE-2.0.html))

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| פישוט של פעולות מסובכות | קוד פתוח – אין אחריות לאמינות הקוד |
| תמיכה במספר רב של דפדפנים |  |
| ערך מוסף (כפי שהוזכר ברשימת התכונות) |  |

* 1. כתיבת תסריט:

חלק מרכזי בפרוייקט הוא הרצת תסריט פעולה אשר הזין המשתמש. נבחנו מספר דרכים שונות להזין ולנתח תסריטים, הן מבחינת השימושיות למשתמש והן מבחינת עקרון הפעולה (השפה) של התסריט:

* + 1. שפת תסריט סטנדרטית (Python)

Python היא שפת תסריט מונחית עצמים, בעלת יכולות נרחבות, ומשמשת כשפת תסריט במגוון אפליקציות. ניתן לשלב אותה באפליקציות NET. באמצעות מודול הנקרא IronPython [/http://ironpython.net] המאפשר גישה של תסריטי Python ל-Assemblies של NET.

לדוגמה:

so=SiteObject('q')

if so.value == "expected value":

so.type("new value")

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| שימוש בשפה סטנדרטית ומוכרת | נדרש מהמשתמש ידע בתכנות |
| ניתן לממש תסריטים מסובכים בזכות עושר השפה | לא מאובטח מספיק(ניתן לגשת באמצעות התסריט לכל אובייקט שקיים באפליקציה) |

* + 1. שפת תסריט ייעודית

נבדק הרעיון לממש מנתח תחבירי ייעודי לאפליקציה (Parser), כך שהוא יהיה מותאם לעולם המושגים ולאובייקטים של האפליקציה. המנתח התחבירי יקבל מחרוזת טקסט, יינתח אותה וימיר אותה לאובייקטי תסריט

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| שימוש בשפה ייעודית ליישום ותפורה לצרכיו | נדרש מהמשתמש ללמוד שפה חדשה |
| ניתן לממש תסריטים מסובכים בזכות עושר השפה | מורכבות וסיבוכיות גדולות במימוש מהדר/מפרש לשפה. |
| מאובטח מספיק – השפה תאפשר גישה רק לאובייקטים של האפליקציה |  |

* + 1. בניית תסריט בצורה גרפית

בניית מסכי ממשק באפליקציה באמצעותם ניתן יהיה לבנות תסריט באמצעות בחירה והכנסת טקסט מינימלית.(ראה SDD)

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| לא נדרש מהמשתמש ללמוד שפת תכנות חדשה | שיטה זו מוגבלת בעושר התסריטים אותם ניתן לכתוב מכיוון שהבחירה היא מוגבלת |
| יחסית פשוט למימוש |  |
| מאובטח מספיק – הממשק יאפשר גישה רק לאובייקטים של האפליקציה |  |
| התסריט נשמר בסיס הנתונים |  |

* 1. בסיס נתונים
     1. MS SQL Compact 3.5

בסיס נתונים מבוסס MS Sql Server אבל בעל תכונות וקיבולת מוגבלת.

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| קיים ממשק ב-VS | מוגבל (אין שרת ולכן אין יכולות כגון Stored Procedures) |
| חינמי |  |
|  |  |
|  |  |

* + 1. Sqlite

טבלה

|  |  |
| --- | --- |
| **יתרונות** | **חסרונות** |
| קיים ממשק ב-VS(Pro) (צריך התקנה מיוחדת) | מוגבל (אין שרת ולכן אין יכולות כגון Stored Procedures) |
| חינמי |  |
| חתימת זיכרון נמוכה |  |
|  |  |

* 1. סוג דפדפן:
     1. IE(stand alone)
     2. Firefox(stand alone)
     3. Embedded IE

הכוונה כאן היא לפקד IE המוטמע בתוך טופס.

1. הפתרון שנבחר לתיכון המערכת

הפתרון שנבחר לתיכון המערכת יתבצע על סמך מתודולוגית OOD, כאשר ישנה חלוקת אחריות למספר שכבות פונקציונליות:

* DAL – Data Access Layer
* Entity Layer
* BLL – Business Logic Layer
* UI – User Interface

ניתן לבחון את חלוקת האחריות והפונקציונליות בכל שכבה, וכמו כן להעמיק לפרטי תתי הרכיבים של המערכת בנספח .

1. תוכנית בדיקות תוכנה

תוכנית בדיקות התוכנה כוללת פירוט מלא של רכיבי המערכת לבדיקה בכל שלבי הפרוייקט, מתודולוגיות הבדיקה וסביבת הבדיקות הנדרשת. כל זאת מתואר בפירוט בנספח STD – Software Test Description

1. לוחות זמנים מעודכנים להמשך העבודה והערכת ביצוע המשימות.

ניתן לצפות בלוחות הזמנים בנספח .

1. ניהול סיכונים מעודכן

תוכנית ניהול הסיכונים מופיעה בנספח

1. רשימת מקורות

אין מקורות נוספים לציין.

1. נספחים
   1. SRD – Software Requirements Document

**1 Introduction**

**1.1 Purpose.**

The purpose of this document is to describe the requirement needed for the Automatic Web Agent System system.

The document is divided to sections, where as each section will describe the requirements from a single module.

**1.2 Scope of the software**

Nowadays web use has become common to practically every household. The web enables to perform many diverse tasks, ranging from acquiring entertainment data (movies, songs, gossip etc…) to financial transactions.

Most of the tasks a user might perform, require the user to authenticate his identity, track data on the website and perform actions based on knowledge acquired on other websites. These actions are very similar in most of the tasks regardless of their nature, all of which require the attention of the user and might be extremely time consuming.

The Automatic Web Agent application project was devised to relieve the average/advanced user from these routine and time consuming tasks.

The Automatic Web Agent will enable the user to compose a script of actions to be taken on specific websites according to data sampled on other sites and conditions that the users will set, and determine the scheduling of the script execution.

For example, there can be a script that will sample the users favorite sky website every 4 hours and will monitor the snow depth of the sky track. Once a preset depth is detected the script will extract the prices of plane tickets to the sky track and email them to the user.

Another example can be a script that will extend the lending period of a book in the Afeka college library, which is not equipped today with a built-in agent, once every 3 days.

Alternatively, an example from the business sector could be a for a company which sells products via the web, a script can track the competitors web sites for price changes and update the prices of the products accordingly.

**1.3 Definitions, acronyms and abbreviations.**

|  |  |
| --- | --- |
| DOM | Document Object Model |
| WWW | World Wide Web |
| DB | Data Base |
| Site Object | a C# Object that represent a specific element in the DOM of the website |
| HTML | Hyper Text Markup Language |
| DHTML | Dynamic HTML |
| URL | Universal Resource Link |
| HTTP | Hyper Text Transfer Protocol |
| SQL | Simple Query Language |
| API | Application Programming Interface |
| CSS | Cascading Style Sheet |

Table

**1.4 References**.

1. Jon Kleinberg. **Authoritative sources in a hyperlinked environment**. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. [[pdf]](http://www.cs.cornell.edu/home/kleinber/auth.pdf)
2. Soumen Chakrabarti, Martin van den Berg, Byron Dom, **Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery**, WWW8 [[html]](http://www8.org/w8-papers/5a-search-query/crawling/index.html)
3. Steve Lawrence and Lee Giles, **Searching the World Wide Web,** Science 1998 [[pdf]](http://www.neci.nj.nec.com/~lawrence/papers/search-science98/search-science98.pdf)
4. Jeffrey Dean, Monika R. Henzinger, **Finding Related Pages in the World Wide Web**, WWW8 [[html]](http://www8.org/w8-papers/4a-search-mining/finding/finding.html)
5. Oren Zamir and Oren Etzioni, **Grouper: A Dynamic Clustering Interface to Web Search Results**, WWW8, [[html]](http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html)
6. Rajan M. Lukose and Bernardo A. Huberman, **Surfing as a Real Option** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/options.html)
7. Bernardo A. Huberman and Rajan M. Lukose, **Social Dilemmas and Internet Congestion** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/storms.htm)
8. Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan M. Lukose, **Strong Regularities in World Wide Web Surfing**, Nature ??? [[abstact]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/surfing.html)
9. William W. Cohen, Wei Fan, **Web-Collaborative Filtering: Recommending Music By Spidering the Web**, WWW9 [[html]](http://www9.org/w9cdrom/266/266.html)
10. Mike Perkowitz Oren Etzioni, **Towards Adaptive Web Sites: Conceptual Framework and Case Study**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/towards/towards.html)
11. Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki, **Unintrusive Customization Techniques for Web Advertising**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/unintrusive/unintrusive.html)
12. Jim Hendler, **Is there an intelligent agent in your future?** Nature, Web matters, 11 March 1999. [[html]](http://helix.nature.com/webmatters/agents/agents.html)
13. Steve Kirsch, **The future of Internet search (keynote address)**, SIGIR 1999 [[pdf]](http://www.acm.org/pubs/articles/proceedings/ir/290941/p1-kirsch/p1-kirsch.pdf)
14. G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure, Proceedings of the 8th International Computing and Combinatorics Conference (COCOON), 2002.
15. [G. Pandurangan](http://www.cs.brown.edu/people/gopal) , [P. Raghavan](http://theory.stanford.edu/people/raghavan/index.html) , and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Building Low-Diameter P2P Networks**](http://www.cs.brown.edu/research/webagent/focs-2001.pdf). Proceedings of the 42th IEEE Symp. on Foundations of Computer Science. 2001.
16. S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**The Web as a graph**](http://www.cs.brown.edu/research/webagent/pods-2000.pdf). Proceedings of the 19th ACM Symposium on Principles of Database Systems, pp 1-10, 2000.
17. Thomas Hofmann, [**Learning Probabilistic Models of the Web**](http://www.cs.brown.edu/research/webagent/Hofmann-SIGIR00.pdf), ACM SIGIR 2000
18. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Stochastic models for the Web graph**](http://www.cs.brown.edu/research/webagent/focs-2000.pdf). Proceedings of the 41th IEEE Symp. on Foundations of Computer Science. 2000.

1. <http://en.wikipedia.org/wiki/Trident_%28layout_engine%29> , downloaded 1/9/2009
2. <http://msdn.microsoft.com/en-us/magazine/cc163723.aspx> , downloaded 1/9/2009

**1.5 Overview of the document.**

This document is mainly aimed for the developers and technical staff involved in the project, and will provide a basic understanding of the requirements to be met on project completion. The main points of interest for a developer would be:

1. The functional requirement listing, in section 3, interface requirements and the security and performance requirements near the end of that section.

2. The model description in section 2.7.

For the IT personnel, the primary points of interest are:

1. Environment considerations – section 2.4

2. Resource requirements – sections 3.4, and

3. Security requirements – section 3.6

**2 General Descriptions**

**2.1 Relation to current projects.**

This project is somewhat dependent on the open source Web Test API called WatiN [[1]](#footnote-1) the current version used (WatiN-2.0.10.928-net-2.0) should be enough for this project.

**2.2 Relation to predecessor and successor projects.**

NA – this is a new project.

**2.3 Function and purpose**

|  |  |
| --- | --- |
| Function | purpose |
| DOM element recording | allow a method for the user to populate the DB with mappings between DOM elements and Site Objects |
| Site Objects management | allow user to change recognition parameters of a DOM element contained in a Site Object |
| Script composition and validation | Allow the user to create the automation script and validate it. |
| Script management | Allow the user to rename, save or delete scripts |
| Script scheduling | Allow the user to schedule the execution time and frequency of the script. |
| Script Execution | Actual execution of the actions in the scripts which include monitoring and manipulating websites and notifications to the user. |

Table

**2.4 Environmental considerations.**

The product will be used on the customer PC as a managed windows application.

The target audience of this product is intermediate to advanced browser user with beginner knowledge in scripting concepts.

The product is designed to run on:

* X86 architecture
* 1GB RAM / HD space TBD
* windows XP OS.

**2.5 Relation to other systems.**

Any website is hosted on a hosting server (HTTP server)

HTTP Server

Auto Web Agent application

WWW

Context diagram 1

When a browser navigates to a website it actually sends a request for a page from the HTTP server (via the URL entry in the browser which translates to a HTTP request). The server then sends the html document to the browser which constructs from it a DOM tree and then renders it to the screen.

In this project the browser is embedded in the Auto Web Agent application.

**2.6 General constraints.**

* Only pure DHTML web sites will be supported (i.e. no flash/flex/SilverLight/activex support).
* The project provided is intended as a proof-of-concept only
* The product will be graphically designed to some extent but the development effort will be put into functionality and features as a priority.

**2.7 Model description.**

The following diagrams will describe the data flow between the various processes in the system.

**2.7.1 DFD Diagrams**

Context Diagram (DFD0):

The system consists of 2 main processes: acquiring data to help recognize the DOM elements of the website and a script composition/execution process



diagram - context diagram

Element Mapping:

The user selects elements in the browser control The elements are then saved in the DB.



diagram – element mapping

Edit element map:

The user (script programmer) selects and marks the priority of the DOM element attribute according to which the element will be identified in the script.



diagram - edit element map

Edit Script:

In this process script is being composed and stored in the DB



diagram – edit script

Script run:

DOM element mapped to site objects are being monitored and manipulated according to the script commands (steps)



diagram – script run

**3 Specific Requirements**

**3.1 Functional requirements.**

* The system shall save all its data in a DB based on MS SQL sdf format (see diagram 6):
  + Element mapping between application objects and DOM elements. This will be done utilizing the website/element/recognition tables where a match of several site object properties will be used to select the corresponding DOM element
  + Script elements that compose scripts, which consist of script/step/condition/action tables

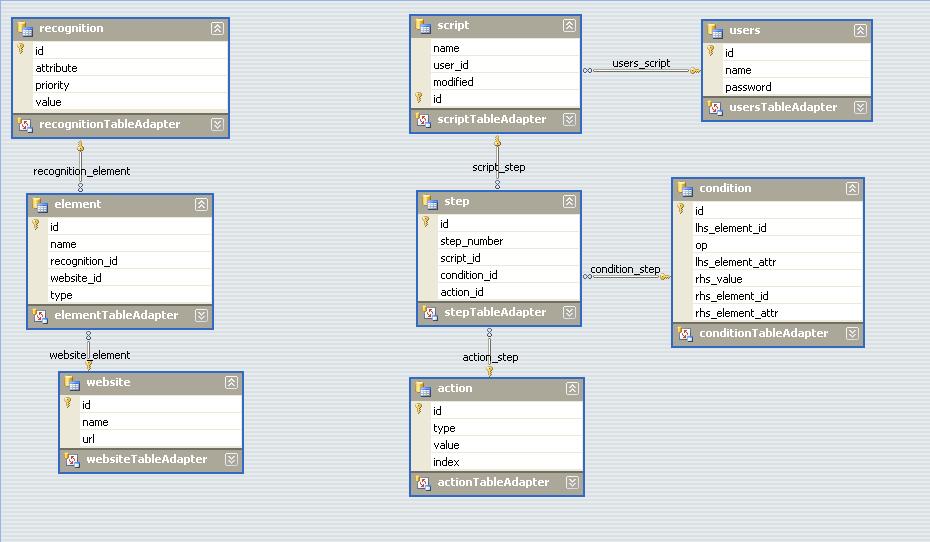


Diagram - ERD

* The application shall have 4 main forms:
  + Browser control form via which the user will select the relevant elements
  + Map editing form where the user will tweak the table that will help in recognizing DOM elements. The table will map between a site object and DOM element attributes.
  + Script scheduling form where the user will determine the time and frequency of script execution.
  + Script execution status and log.
* Element selection will be done using mouse hover over the element and pressing the middle mouse button for confirmation. When the mouse hovers over an element it shall be surrounded by a red rectangle.
* Script execution shall reuse the browser control form.
* Log of execution progress shall be displayed and written to a file.
* Workflow shall be implemented according to the DFD diagrams in section 2.7.1

**3.2 Performance requirements.**

* The system will support at least 3 simultaneous website interactions.
* Element interaction (click/text fill) shall be done in 1 second max.

**3.3 Interface requirements.**

NA

**3.4 Resource requirements.**

* X86 based CPU
* MS windows XP or greater
* 2 GB memory
* 20MB hard disk space

**3.5 Verification requirements.** As in test plan document

**3.6 Security requirements.**

All site and user data shall be stored in a password protected DB file.

* 1. SDD – Software Design Description

1. Introduction.
   1. **Purpose.**

The purpose of this document is to specify all aspects of the software system design for the "Automatic Web Agent" Project.

The main audience for this document is developers, basing system implementation on the application interfaces and functional flows laid out in this document.

This document will also provide sections aiding management or external consulting staff by describing system architecture, referenced components and modules and system interfaces, by which a general understanding of the system may be attained.

* 1. **Definitions, acronyms and abbreviations.**

Table

|  |  |
| --- | --- |
| DOM | Document Object Model |
| WWW | World Wide Web |
| DB | Data Base |
| Site Object | a C# Object that represent a specific element in the DOM of the website |
| HTML | Hyper Text Markup Language |
| DHTML | Dynamic HTML |
| URL | Universal Resource Link |
| HTTP | Hyper Text Transfer Protocol |
| SQL | Simple Query Language |
| API | Application Programming Interface |
| CSS | Cascading Style Sheet |

* 1. **References.** May extend/delete information from SRD Section 1.4.

1. Jon Kleinberg. **Authoritative sources in a hyperlinked environment**. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. [[pdf]](http://www.cs.cornell.edu/home/kleinber/auth.pdf)
2. Soumen Chakrabarti, Martin van den Berg, Byron Dom, **Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery**, WWW8 [[html]](http://www8.org/w8-papers/5a-search-query/crawling/index.html)
3. Steve Lawrence and Lee Giles, **Searching the World Wide Web,** Science 1998 [[pdf]](http://www.neci.nj.nec.com/~lawrence/papers/search-science98/search-science98.pdf)
4. Jeffrey Dean, Monika R. Henzinger, **Finding Related Pages in the World Wide Web**, WWW8 [[html]](http://www8.org/w8-papers/4a-search-mining/finding/finding.html)
5. Oren Zamir and Oren Etzioni, **Grouper: A Dynamic Clustering Interface to Web Search Results**, WWW8, [[html]](http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html)
6. Rajan M. Lukose and Bernardo A. Huberman, **Surfing as a Real Option** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/options.html)
7. Bernardo A. Huberman and Rajan M. Lukose, **Social Dilemmas and Internet Congestion** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/storms.htm)
8. Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan M. Lukose, **Strong Regularities in World Wide Web Surfing**, Nature ??? [[abstact]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/surfing.html)
9. William W. Cohen, Wei Fan, **Web-Collaborative Filtering: Recommending Music By Spidering the Web**, WWW9 [[html]](http://www9.org/w9cdrom/266/266.html)
10. Mike Perkowitz Oren Etzioni, **Towards Adaptive Web Sites: Conceptual Framework and Case Study**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/towards/towards.html)
11. Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki, **Unintrusive Customization Techniques for Web Advertising**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/unintrusive/unintrusive.html)
12. Jim Hendler, **Is there an intelligent agent in your future?** Nature, Web matters, 11 March 1999. [[html]](http://helix.nature.com/webmatters/agents/agents.html)
13. Steve Kirsch, **The future of Internet search (keynote address)**, SIGIR 1999 [[pdf]](http://www.acm.org/pubs/articles/proceedings/ir/290941/p1-kirsch/p1-kirsch.pdf)
14. G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure, Proceedings of the 8th International Computing and Combinatorics Conference (COCOON), 2002.
15. [G. Pandurangan](http://www.cs.brown.edu/people/gopal) , [P. Raghavan](http://theory.stanford.edu/people/raghavan/index.html) , and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Building Low-Diameter P2P Networks**](http://www.cs.brown.edu/research/webagent/focs-2001.pdf). Proceedings of the 42th IEEE Symp. on Foundations of Computer Science. 2001.
16. S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**The Web as a graph**](http://www.cs.brown.edu/research/webagent/pods-2000.pdf). Proceedings of the 19th ACM Symposium on Principles of Database Systems, pp 1-10, 2000.
17. Thomas Hofmann, [**Learning Probabilistic Models of the Web**](http://www.cs.brown.edu/research/webagent/Hofmann-SIGIR00.pdf), ACM SIGIR 2000
18. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Stochastic models for the Web graph**](http://www.cs.brown.edu/research/webagent/focs-2000.pdf). Proceedings of the 41th IEEE Symp. on Foundations of Computer Science. 2000.
19. <http://en.wikipedia.org/wiki/Trident_%28layout_engine%29> , downloaded 1/9/2009
20. <http://msdn.microsoft.com/en-us/magazine/cc163723.aspx> , downloaded 1/9/2009
21. System Architectural Design
    1. System Architecture

The system architecture is composed of 3 main modules:

* Persistency Module
* Mapping module
* Scripting Module

The Persistency module contains the database, and the components to manipulate and query it's data (Data Access Layer). The DB is responsible to persist element mappings and scripting information.

The Mapping Module is composed of UI in which the user can map web page elements to scriptable objects (also known as SiteObjects), which are saved in the DB.

The scripting module is composed of a UI in which the user can create a script to manipulate and monitor a website pages displayed in IE. This module is responsible to schedule the script run time and to notify the user of various events.

The following drawing illustrates the high level architecture which will be discussed in detail in section .

illustration

Data Access Layer

DB

Persistency Module

Embedded Browser

DOM Analysis

Mapping Control

Mapping Module

Script Builder

Script Scheduler

Script Logger

Script Runner

Scripting Module

The data flow diagram can be found in the SRD appendix.

* 1. System Interface Description.

The system has a single external interface which is the windows GUI, described is section

1. Detailed Description Of Components. Gives detailed component information.

**3.n** **[Component identifier]** Fill in name of component here.

**3.n.1 Type**. Could be a module, an input/output/temporary file, a program, a class, a script, a web page, etc.  
**3.n.2 Purpose**. Describe the purpose of the component.  
**3.n.3 Function Inputs and Outputs**. Describe the functionality of the component, including its interface properties (call and return types) and logical behavior.  
**3.n.6 Interfaces.** Define the control and data flow to and from the object. Gives a detailed picture of its context in the overall system architecture. **3.n.7 Data.** Describe in detail (where possible) the data values and data structures belonging to this component. Otherwise give an outline description.

* 1. Database
     1. Type: module
     2. Purpose

The type of the database is sdf (Compact relational database developed by Microsoft, also known the SQL Server Compact format; designed for applications that run on mobile devices and desktops; contains the complete database contents and can be up to 4GB in size.)

The database persist several functions:

* Users and password
* Mappings between DOM elements and SiteObject elements.
* Scripts
  + 1. Function Input And Outputs

All functions working on the database are in the DAL class.

* + 1. Interfaces

In order to connect to the database a data access object is required. The system uses a dataset (AutoWebAgentDBDataSet.xsd) to map the database to objects.

The dataset contains a table adapter object for each table in the database and through it allows read/write access to the database.

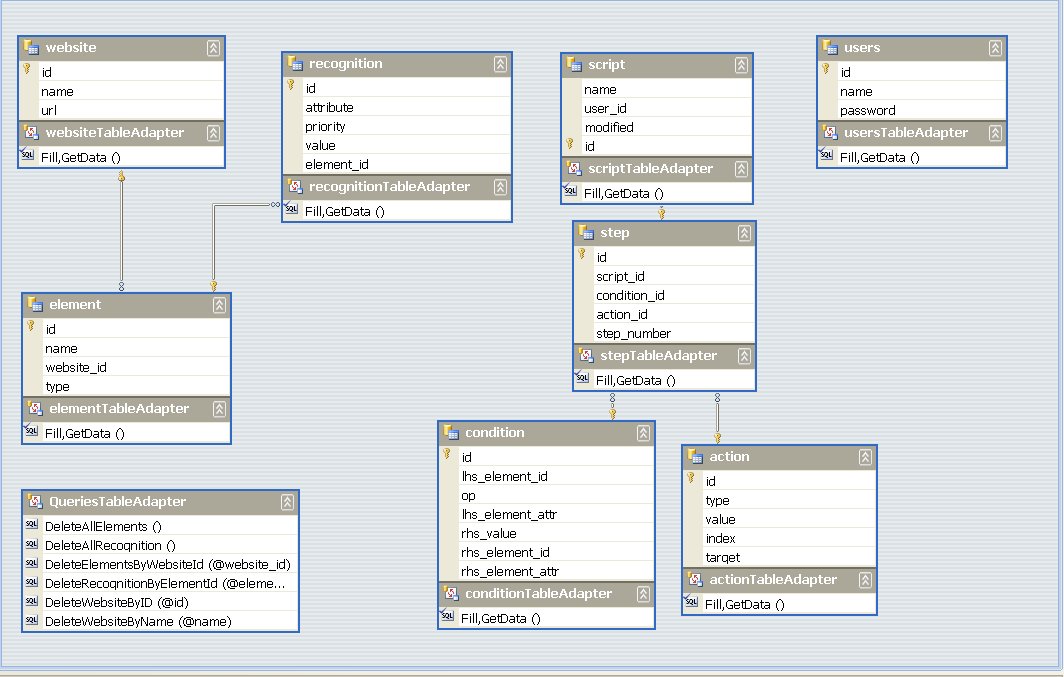
The data in the file is encrypted and in order to decrypt it a connection string must be used with the format:

Data Source=<path to .sdf file>;Password=<password>;Persist Security Info=True

* + 1. Data

The database contains the tables in and their corresponding adapters. The QueriesTableAdaper is used to have custom operation on the tables.

illustration



* 1. Data Access Layer (DAL)
     1. Type: Class
     2. Purpose

Provide a simplified access to the database, such that each table will be represented by an object, and also will contain methods to save long and complicated SQL sentences.

* + 1. Function Input And Outputs
* public IEnumerable<RecognitionProperty> GetElementRecogitionProperties(int wid, int eid): this function gets the website id and element id and returns a list of matching element recognition properties.
* public IEnumerable<RecognitionProperty> GetElementRecogitionProperties(string wName, string eName): this finction gets a website name and an element name and returns a list of matching element recognition properties.
* public int GetWebsiteID(string url): this function gets a website url and return its id in the DB.
* public int GetWebsiteIDByName(string name): this function gets a website id by searching it's name.
* public string GetWebSiteUrlByName(string wName): this function get a website id by its URL.
* public AutoWebAgentDBDataSet.websiteRow GetWebsiteRow(string url): this function gets a site URL and returns its matching rows in the website table.
* public AutoWebAgentDBDataSet.websiteRow GetWebsiteRow(string name,out bool found): this function gets a site name and returns its matching rows in the website table.
* public void SaveChanges(): this function commits changes to the DB.
* public int SaveChanges(params string[] tableNames): this function commits changes to the given tables
* public DataTable CreateElementRecognitionView(int WebsiteID): this function gets a website id and returns a DataTable for the Web Recorder View.
* private void CreateDB(string name): this function gets a DB name and creates a new DB.
* public int SetElement(AutoWebAgentDBDataSet.websiteRow WebsiteRow, string elementName, string elementType, List<RecognitionProperty> elementProperties ): this function gets a website row, an element name, an element type, and a list of recognition properties update the proper tables and return a 1 for success.
* public void ClearWebsiteData(int wid): this function gets a website id and deletes all entries relating to it.
  + 1. Interfaces

The mapping or scripting module may take an instance of this class in order to gain access to the DB.

* + 1. Data

The DAL::DB property holds an instance of awaDAL.AutoWebAgentDBDataSet dataset, which allows access to the DB.

* 1. Mapping Control
     1. Type: Method
     2. Purpose

The purpose of this module is to translate the element selected by the user in the GUI to element mapping in the data base (this maps a name of a DOM element and a set of recognition properties that will help identify it when the script runs and search this element in a page).

* + 1. Function Input And Outputs

The method implementing the function is contained in the webRecorderForm class and is called private void saveElementButton\_Click(object sender, EventArgs e)

It has the standard "event raised" arguments (sender object, and EventArgs object) and return no value.

* + 1. Interfaces

Each time the user presses the 'apply' button on the recorder GUI this method is called. If the user entered an element name it will take it otherwise it will generate a name if possible, then it will gather the recognition properties of the selected element in a list and map between the list and the name using the DAL::SetElement method described in

* + 1. Data

The method holds an instance of List<DAL.RecognitionProperty> list which contains RecognitionProperty entries.

* 1. Embedded Browser
     1. Type: module
     2. Purpose

The embedded browser is a Microsoft provided WebBrowser Control found in System.Windows.Forms assembly. It is part of the web recording form GUI and is used for helping the user select the element he wants to map. When the user hovers the mouse over an element it is surrounded by a red rectangle. When the user double click the SHIFT key the surrounded element is sent to the DOM analyzer to extract it's recognition properties.

* + 1. Function Input And Outputs

The following function are given to their respective events in the web browser control:

* void webBrowser\_DocumentCompleted(object sender, WebBrowserDocumentCompletedEventArgs e)
* void Document\_MouseOver(object sender, HtmlElementEventArgs e)
* void webBrowser\_DocumentCompleted(object sender, WebBrowserDocumentCompletedEventArgs e)
* private void webBrowser\_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
  + 1. Interfaces

The web browser control exposes a Document object to interact with the page and also exposes the following events which are used by the web recorder form:

* DocumentCompleted – Is raised when the browser finished loading a page.
* Document.MouseOver – Is raised when the mouse hovers an element
* Document.MouseLeave – Is raised when the mouse leaves an element.
* PreviewKeyDown – Is raised when a key is pressed on the keyboard (before processing it)
  + 1. Data

The only relevant data the web browser control holds is the Document object which is used to interact with the page's DOM.

* 1. DOM Analysis
     1. Type: method
     2. Purpose

The purpose of the DOM analysis method is to analyze the selected DOM element and extract from recognition properties, such that it will be recognized uniquely and could be search and found by these properties at script runtime.

Recognition properties can be attributes of the element (for example 'id' or 'name') or calculated path to the element from another know element which is called 'indirectPath' in the case the element has no 'id' or 'name' attributes.

The algorithm to calculate the 'indirectPath' is as follows:

*if element has no id/name*

*1. determined position in parent children by counting the*

*depth of nextSibling loop*

*2. push count to stack*

*3. then check if parent have id attribute: if true then push*

*id to stack else if parent=null stop else repeat 1-3*

*4. serialize stack content and create a fake attribute:*

*indirectPath => [stack contents] with the format:*

*id:<id>|<name:<name>> <children indexes seperated by space>*

* + 1. Function Input And Outputs

The method signature is:

private void analyzeElement(HtmlElement currentElement)

it gets an HtmlElement object which represents the element the user selected, and doesn't return any value.

* + 1. Interfaces

This method is called when the user hovers over an html element on a page and the presses the SHIFT key twice in a row.

* + 1. Data

The method holds an instance of List<DAL.RecognitionProperty> list which contains RecognitionProperty entries.

* 1. Script Builder
     1. Type: Class
     2. Purpose

The purpose of this class is to enable the user to visually build a browser automation script.

The process of building a script is composed of several steps (terms are taken from the GUI):

* Add a script and set its name in a 'add script' dialog
* Add steps to the script, each step will require the following:
  + Determine the step name in the 'Add step' dialog.
  + Set the conditions to activate the step. the condition types can be:
    - True – always perform the actions.
    - False – never perform the actions.
    - Value – perform the actions only if source attribute is equal to 'Value'.
    - Equal – perform the actions only of source attribute equals target attribute
    - Checked – perform the actions only if source has a 'checked' attribute on.
    - Selected – perform the action only if source has the 'selected' attribute on.
  + Set the actions. Action types can be:
    - Select – selects source (valid for list box)
    - Check – checks source (valid for checkbox)
    - Uncheck – uncheck source (valid for checkbox)
    - Type Text – types 'Value' in source
    - Notify – generates a notification using the selected method(email, log, popup)
    1. Function Inputs and Outputs
    2. Interfaces
    3. Data
  1. Script Scheduler
     1. Type: Class
     2. Purpose

The purpose of this class is to set the scheduling parameters of a script. A script may run once, at a designated time, or run periodically in an assigned time span. The period time is set by the user, which select the amount of time and the unit for each period.

* + 1. Function Inputs and Outputs
    2. Interfaces
    3. Data
  1. Script Runner
     1. Type: Class
     2. Purpose

The purpose of this class is to enable the user select which of the scripts will be active. Only active scripts will be checked for scheduling and running.

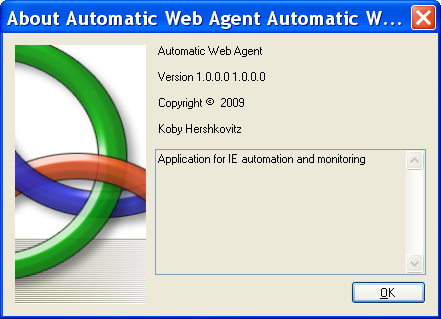
* + 1. Function Inputs and Outputs
    2. Interfaces
    3. Data
  1. Script Logger
     1. Type: class
     2. Purpose

The purpose of the logger class is to log script activations, script runs, script results. The log is written to the log text box and to the windows event log.

* + 1. Function Inputs and Outputs
    2. Interfaces
    3. Data

1. User Interface Design
   1. About Screen
      1. When selecting Help-> About this screen will appear with the copyright details of the application
      2. Screen image

illustration

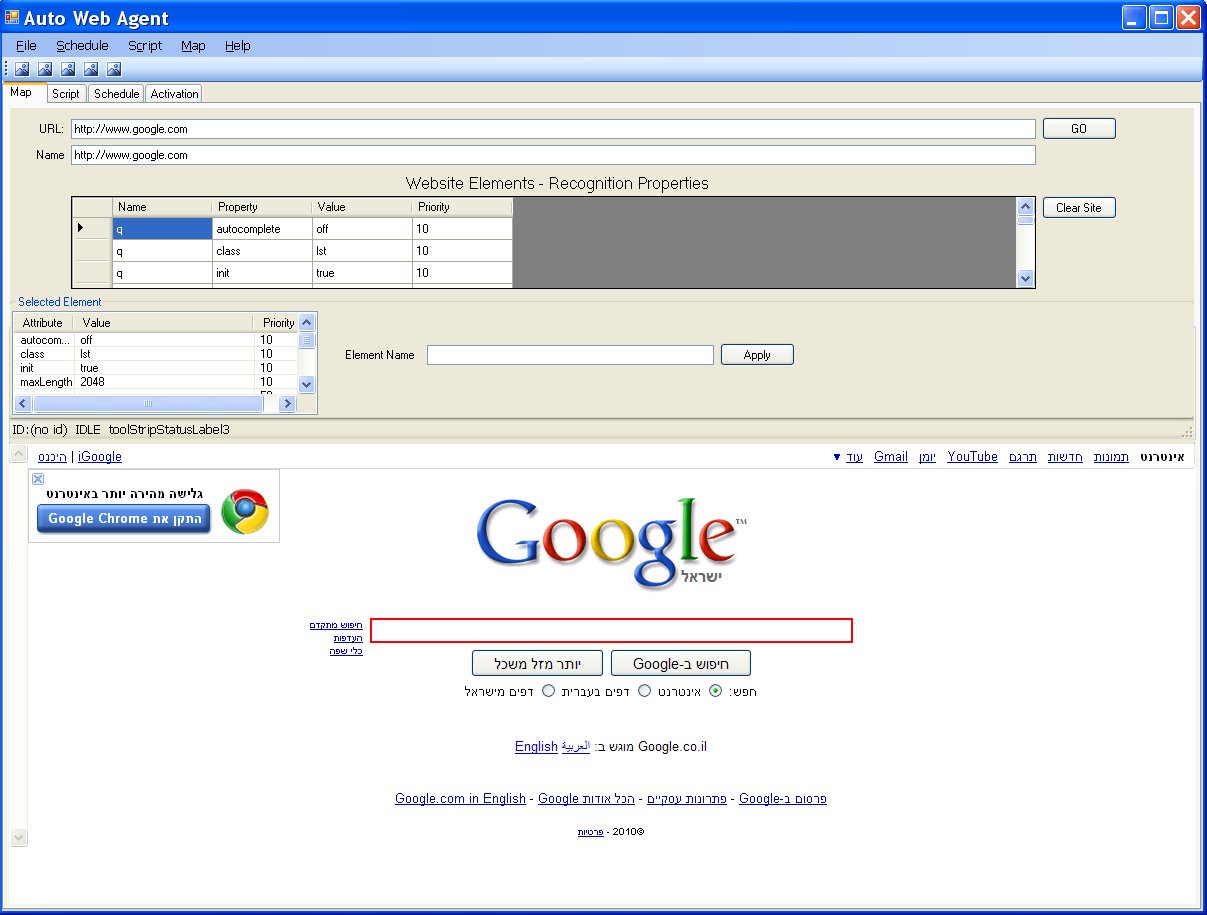


* + 1. Objects and Actions

This is a plain and standard 'About' screen with application name, author, copyright, logo and an 'OK' button to make it go away.

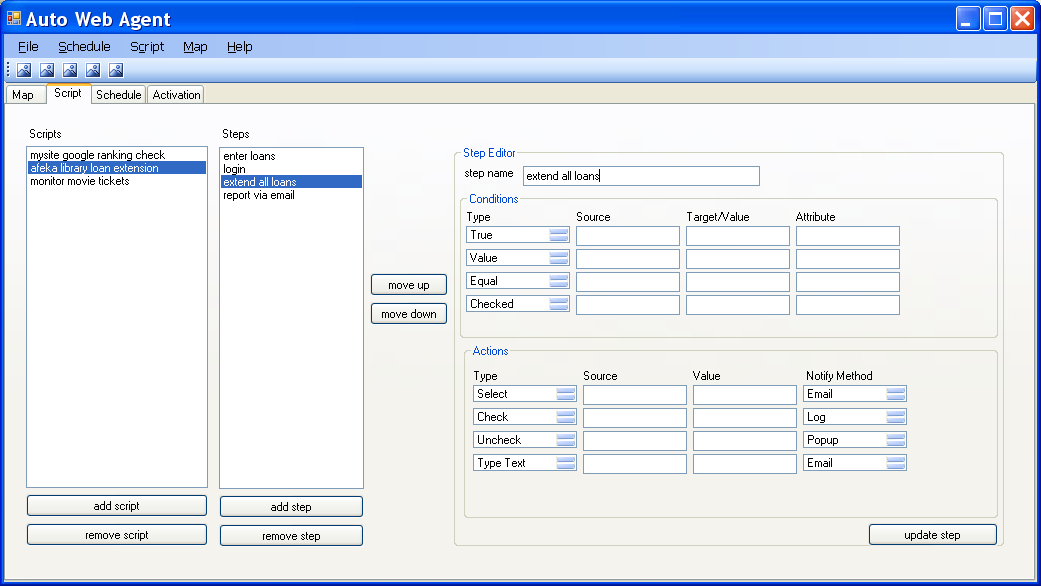
* 1. Mapping Screen
     1. This screen appears when selecting the Map tab
     2. Screen image

illustration



* + 1. Objects and Actions
* URL text box: the user enters the website URL. Pressing enter or the 'Go' button will cause the web browser control to navigate to the URL.
* Name text box: the associated name to the URL. This name will be used to address the website in the scripts. It defaults to the URL if the user leaves it blank.
* Website Elements table: displays the recognition properties for the elements saved in the DB. It contain the following fields:
  + Name: the element's name
  + Property: element property (such as id, name, class etc)
  + Value: property value
  + Priority: priority assigned for recognition algorithm in the scripting module.
* Clear site button: deletes all data related to a website.
* Selected Element table: this table shows the selected element properties. The user may change the values discovered before applying element to the DB.
* Element name textbox: used to assign a user defined name to the selected element. If empty it defaults to the element id. If element id doesn't exist then it defaults to the element name and if if that doesn't exist then an error popup is displayed.
* Apply button: commit selected element to the DB
* Web browser control: used to display the web page and to select elements on it by hovering over an element a red rectangle is displayed around an html element, double pressing the shift key will load its properties to the 'Selected Element' table.
  1. Script Screen
     1. This screen appears when selecting the Script tab
     2. Screen image

illustration

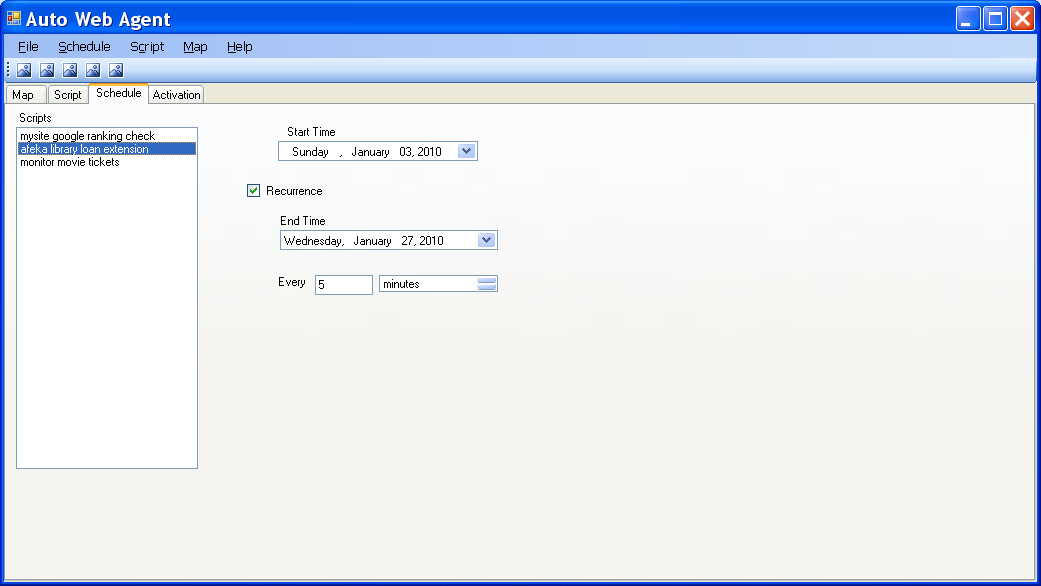


* + 1. Objects and Actions
* Scripts list box: list box that contains all the defined scripts.
* Add script button: opens a dialog which asks for the new script name. After pressing the ok the script is added to the scripts list box.
* Remove script button: delete the selected script along with its steps.
* Steps list box: list box that contain the steps in the selected script.
* Add step button: opens a dialog which asks for the new step name. After pressing the ok the step is added to the steps list box.
* Remove step button: delete the selected step.
* Move up/down buttons: moves up/down the selected step in the Steps list box.
* Step editor displays information about the selected step. A step is composed of 2 parts: the condition part and the action part. Only if the condition part evaluates to true the action part will be executed. The GUI supports 4 conditions and 4 actions per step. The condition part has 4 fields:
  + Type – true, false, equal, value, selected, checked
  + Source – the element name under test
  + Target/value – target is the element name for comparison in 'equal' condition, value is valid for value condition
  + Attribute –the attribute under test

The action part has 4 fields:

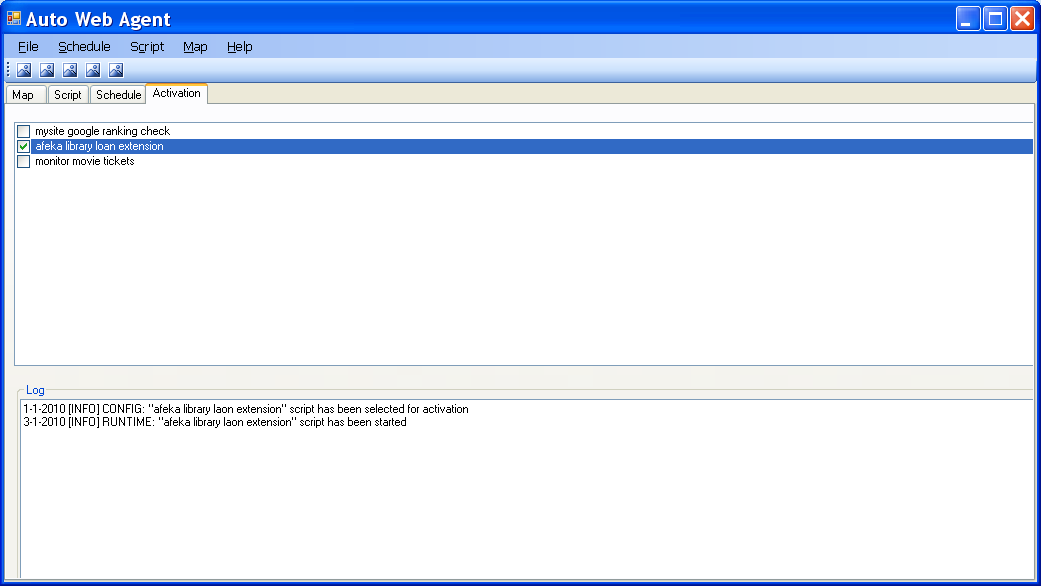
* Type – select, check, uncheck, type text, notify
* Source - the element which will be changed
* Value – the value for a 'type text' or 'select' action
* Notify method – email, log, popup. This field is valid only for the notify action.
* Update step button: commit step to DB.
  1. Schedule Screen
     1. This screen appears when selecting the Schedule tab
     2. Screen Image

illustration



* + 1. Objects and Actions
* Scripts list box: used for the user to select the script for which scheduling parameters will be set.
* Start time picker control: used to select the start time of the script run
* Recurrence checkbox: if checked enables recurrence of script run
* End time picker control: last time the script can run
* Interval text input box: holds the time interval between runs
* Time unit list box: used o select the unit of the interval time.
  1. Activation Screen
     1. This screen appears when selecting the Activation tab
     2. Screen Image

illustration



* + 1. Objects and Actions
* Checked list box: used for selecting script for activation. Only selected scripts will be considered for scheduling/run.
* Log list box: used to log events in the system, like depicted in .

Additional Material

* 1. STD – Software Test Description

1. **Introduction**

This test plan will verify the entire requirements in the Auto Web Agent SRD were implemented correctly. The philosophy of testing is to test each component as isolated as possible (Unit test each component by building a test class for each component class) and then perform tests on the system as a whole.

Three synthetic websites (i.e. html files) with assorted types of elements will be used as the test subjects.

**Test items**

* + - Element processing
    - Script processing

1. **Features to be tested**

* Test functionality of element recognition:
  + Select text input element and verify it is entered in the DB correctly.
  + Select radio button and verify it is entered in the DB correctly.
  + Select a list element and verify it is entered in the DB correctly.
  + Select button and verify it is entered in the DB correctly.
  + Select checkbox and verify it is entered in the DB correctly.
* Test functionality of map editor
  + Change name of several elements and verify they are changed in the DB
  + Change the recognition priorities/attributes of an element and verify it is reflected in the DB
* Test script editor
  + Create new script
  + Add several steps
  + Delete several steps
  + Save script to DB.
* Test script scheduling and execution
  + Schedule a script to run after 10 minutes, check it executes correctly.
  + Schedule a script to run after 1 hour for 3 times, check it executes correctly.
  + Schedule a script to loop indefinitely, check it executes correctly.

1. **Features not to be tested:**
   1. the open source WatiN library.

1. **Environmental needs –** as in project proposal section 4.
2. **Schedule** – as in Gantt chart.
   1. SPMP – Software Project Management Plan
3. Introduction - Project Management Plan Overview

This appendix will describe the progress made and the progress planed of the project execution.

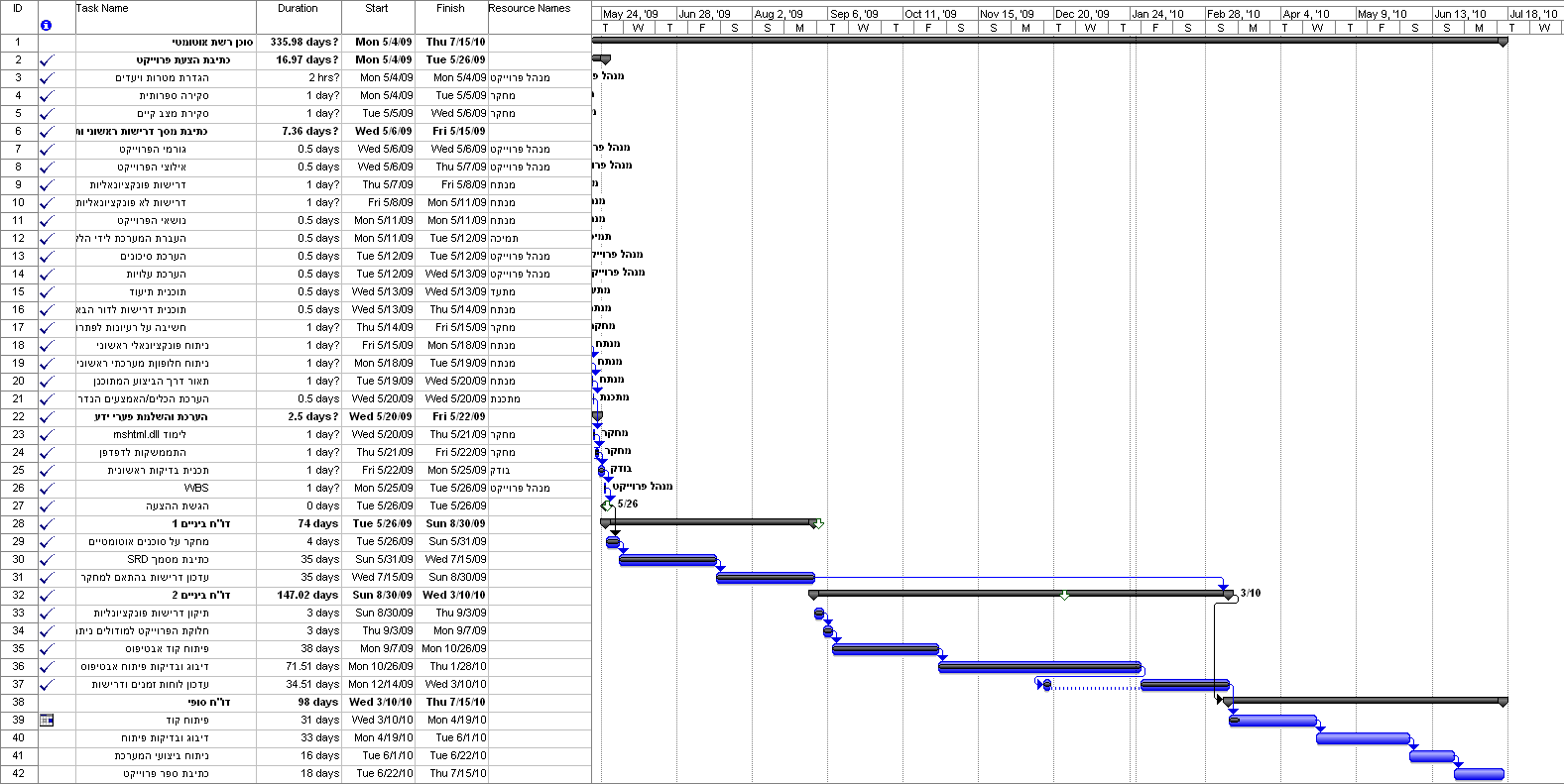
1. Risk Management - Describes updates of risks and managing them.

Due to main developer (me…) birth of new daughter, time tables may be delayed by factor of 3-4 weeks.

1. Evaluation of the SPMP

The progress of the project is more or less on track except for a 2 week delay in deliver the second report. The prototype coding was begun module by module and then integrated into an application. The next thing is to make it robust, complete all features, test it and fix all found bugs.

1. Schedule – Updated and Detailed Work Plan.



* 1. הצעת הפרוייקט

****

**מחלקת הנדסת תוכנה**

שם הפרויקט: סוכן רשת אוטומטי

Project Name: Automatic Web Agent

הצעת פרויקט

|  |  |
| --- | --- |
| שם הסטודנט: | קובי הרשקוביץ |
| מספר תעודת זהות: | 025632654 |
| שם המנחה: | אלעד הוגן |
| תאריך ההגשה: | 25/05/09 |

**תוכן ענינים**

[מבוא 4](#_Toc229036130)

[מטרות, יעדים ומדדים 5](#_Toc229036131)

[סקירה ספרותית 6](#_Toc229036132)

[סקירת מצב קיים 7](#_Toc229036133)

[מסמך דרישות ראשוני ותיחום של המערכת 7](#_Toc229036134)

[גורמי הפרויקט 7](#_Toc229036135)

[תכלית הפרויקט 7](#_Toc229036136)

[לקוחות הפרויקט 7](#_Toc229036137)

[משתמשי הפרויקט 7](#_Toc229036138)

[אילוצי הפרויקט 7](#_Toc229036139)

[אילוצים ראשיים/קריטיים 7](#_Toc229036140)

[הגדרות שמות, קיצורים וראשי תיבות 7](#_Toc229036141)

[עובדות והנחות הקשורות לפרויקט 8](#_Toc229036142)

[דרישות פונקציונאליות 8](#_Toc229036143)

[תחום העבודה 8](#_Toc229036144)

[תחום המוצר 8](#_Toc229036145)

[דרישות מידע ופונקציונאליות 8](#_Toc229036146)

[דרישות לא פונקציונאליות 8](#_Toc229036147)

[דרישות עיצוב 8](#_Toc229036148)

[דרישות שימוש וממשק משתמש 8](#_Toc229036149)

[דרישות ביצועים 8](#_Toc229036150)

[דרישות הפעלה 8](#_Toc229036151)

[דרישות תחזוקה ותמיכה 9](#_Toc229036152)

[דרישות אבטחה 9](#_Toc229036153)

[דרישות תרבותיות ופוליטיות 9](#_Toc229036154)

[דרישות חוקיות 9](#_Toc229036155)

[נושאי הפרוייקט 9](#_Toc229036156)

[נושאים פתוחים 9](#_Toc229036157)

[פתרונות מדף 9](#_Toc229036158)

[בעיות חדשות 9](#_Toc229036159)

[משימות 9](#_Toc229036160)

[העברת המערכת לידי/לניהול הלקוח 9](#_Toc229036161)

[סיכונים 9](#_Toc229036162)

[עלויות 9](#_Toc229036163)

[תיעוד ולימוד משתמשים 10](#_Toc229036164)

[דרישות לדור הבא 10](#_Toc229036165)

[רעיונות לפתרון 10](#_Toc229036166)

[ניתוח פונקציונאלי ראשוני 10](#_Toc229036167)

[ניתוח חלופות מערכתי ראשוני 13](#_Toc229036168)

[תאור דרך הביצוע המתוכנן 13](#_Toc229036169)

[האמצעים/הכלים הנדרשים 13](#_Toc229036170)

[פערי ידע שעל הסטודנט להשלים 14](#_Toc229036171)

[תוצרי הפרויקט 14](#_Toc229036172)

[תכנית בדיקות ראשונית 15](#_Toc229036173)

[ניהול סיכונים 15](#_Toc229036174)

[תוכנית ניהול הפרויקט 16](#_Toc229036175)

[Work Breakdown Structure 16](#_Toc229036176)

[תוכנית עבודה של הפרויקט 17](#_Toc229036177)

[רשימת מקורות: 17](#_Toc229036178)

# מבוא

כיום השימוש באינטרנט נעשה רווח כמעט בכל בית ובאמצעותו ניתן לבצע משימות רבות ומגוונות החל מקבלת מידע בידורי וכלה בפעולות בנקאיות. במרבית מן הפעולות נדרש המשתמש להזדהות, לבצע מעקב אחר נתונים באתר, ולבצע פעולות על סמך ידע שלמד מאתרים אחרים. פעולות אלו במרבית הזמן זהות/דומות בין האתרים וכמו כן ביצוען עלול לדרוש תשומת לב וזמן רב מן המשתמש.

הפרוייקט המוצע הינו סוכן רשת אוטומטי הבא להחליף מטלות שגרתיות וגוזלות זמן של משתמש דפדפן ממוצע/מתקדם. באמצעות המערכת המוצעת יוכל המשתמש לתכנן תסריט של פעולות מסויימות באתרים מסויימים בהתאם למידע הנדגם מאתרים אחרים ולקבוע את זמן ביצוע התסריט.

לדוגמא ניתן לקבוע תסריט שבו נדגם כל 4 שעות אתר סקי שהמשתמש מחבב במיוחד (לדוגמא www.champery.com) , התסריט מכוון לזהות את השדה של גובה השלד באתר וברגע שגובה השלג מגיע ל-110cm התסריט שולף מאתר ה-"דקה ה-90" את מחירי חבילות הנופש לאתר ושולח דוח מפורט עם הנתונים באי-מייל למשתמש.

דוגמא נוספת יכולה להיות תסריט שיאריך את זמן השאלת ספר באתר הספריה של מכללת אפקה ,שאינו מצוייד כיום בסוכן אוטומטי להארכת השאלות, פעם בשלושה ימים.

יחודיות הפרוייקט היא שבניגוד לסוכנים אוטומטיים הנמצאים באתר מסויים ונשלטים מתוך האתר הפרוייקט הנ"ל מציע סוכן שיתוכנת בצד הלקוח ולא בצד השרת, דבר אשר יאפשר לו נוחויות ושליטה מירבית להשגת מטרותיו.

# מטרות, יעדים ומדדים

**מטרות:**

רבות מהפעולות הפיננסיות, הלימודיות והבידוריות נעשות באמצעות דפדפן ברשת האינטרנט:

* רישום לאתרים
* קניה/מכירה של מניות
* הזמנת כרטיסים
* הגשת מטלות
* בחירת קורסים

רבות ממטלות אלו הן מחזוריות, שגרתיות ועלולות להתיש את המשתמש ו/או לגרום לו לבצע שגיאות באופן הכנסת הנתונים.

מטרת הפרוייקט הנגזרת מבעיה זו היא בניית אפליקציה אשר תבצע מטלות שגרתיות וצפויות מראש בצורה אוטומטית במקום משתמש דפדפן אנושי

**יעדים:**

ניתן לפרוט מטרת על זו ל-3 יעדים:

* ניטור אתר בצורה מחזורית
* ביצוע פעולות אוטומטיות בדפדפן
* התראה למשתמש ע"פ בקשתו על אירוע באתר מסויים.

**מדדים:**

המדדים ע"פ תימדד הצלחת הפרוייקט הם:

* אחוז האתרים הנתמכים מתוך עשרת האתרים הראשונים בחיפוש בגוגל יעמוד על 80% בקטגוריות פיננסים, בידור, ספורט (דוגמא לחיפוש יכולה להיות המילה 'פיננסי')
* נוהל כניסה לאתר הספרייה במכללה והארכת השאלה אם קיימים ספרים מושאלים יסתיים בהצלחה.
* נוהל דגימת אתר לשינויים בתדירות של 6 פעמים בדקה יסתיים בהצלחה (ידווח בצורה נכונה על שינוי).
* משתמש חדש יוכל להגדיר תסריט אוטומטי של דגימת 2 אתרים ואינטראקציה ביניהם בתוך כשעה.

# סקירה ספרותית

סוכני רשת הם מערכות תוכנה מורכבות אשר פועלות ברשת האינטרנט העולמית (www), ברשת הארגונית או ברשתות מקומיות. סוכני רשת מתוכננים לבצע מגוון משימות החל מאגירה וניתוב של מידע וכלה בחיפוש קיטלוג וסינון מידע.

הספרות מתארת מסגרת לתכנון וניתוח של סוכני רשת ומערכות סוכנים בהתבסס על מודלים מתמטיים של הסביבה בה הם פועלים. יש 3 אבני בניין מרכזיות לבניית סוכני רשת:

* מודלים סטוקאסטיים של גרף הרשת אשר משקללים את הפילוג והקישוריות של דפי רשת ונותנים קוים מנחים מרכזיים בתכנון סוכנים, ע"י הפקת תכונות ייחודיות של הסביבה בה הסוכנים פועלים.

הרשת יכולה להיראות כגרף מכוון אשר הצמתים בו הם דפי HTML סטטיים והקשתות בו הם קישורים מאתר אחד למשנהו. מאמר [ ‎18] מציע ומנתח מודלי גרפיים אקראיים שנוצרו ע"י תצפיות נסיוניות ברשת. המודלים שונים מהמודלים המקובלים בכך ש קשתות שנבחרו בצורה בלתי תלויה לא משתקפות בסטטיסטיקה שנצפתה ברשת, כך שקשתות במודלים אלו הינם תלויות סטטיסטית ביניהן. כמו כן נוספים צמתים לגרף כתלות בזמן. דבר זה משקף את העובדה שהרשת משתנה עם הזמן.

* שיטות למידה סטטיסטית המאפשרות לסוכנים ללמוד על סביבתם ע"י הפקת מודלים סטוקאסטיים של דף רשת ומבנה קישורים מקומיים. מאמר [ ] מתאר את הרשת כעולם שבו המוני קישורים מחברים בין מסמכי ודפי אתרים כך שנוצר גרף בעל סיבוכיות גבוהה וללא תקדים וזהו גרף הרשת. מאמר זה מציג גישה חדשה ללימוד מודלים הסתברותיים שך הרשת אשר יכולים לשמש כחלק בתהליך חיזוי קישוריות ומידע של מסמכי רשת. השיטה המתוארת במאמר זה היא שיטה הסתברותית לצמצום מימדים (probabilistic dimension reduction) אשר גוזרת ומאחדת את שיטת הניתוח הסמנטי המאוחר (Latent Semantic Analysis) ואת אלגוריתם קליינברג (Hubs-and-Authorities) בסביבה סטטיסטית.

דבר זה אמור להיות הצעד הראשון לפיתוח תשתית סטטיסטית של טכנולוגיות מידע ברשת. למרות שמאמר זה לא מתמקד ביישום מסויים, מגוון אלגוריתמים הפועלים בסביבת רשת יכולים לנצל את היתרונות בטכניקות המוצגות למטרות שונות כגון מנועי חיפוש, "זוחלי רשת" (Web crawlers) ומערכות סוכני מידע.

* אלגוריתמים לתכנון אוטונומי וקבלת החלטות בסביבת רשת לאפשר לסוכנים להשיג מטרות ולהסתגל לסביבה משתנה [16]

ניקח לדוגמא בעיות קלאסיות בתחום בסיסי נתונים/איחזור מידע כגון חיפוש בטקסט, כריית מידע וסיווג. הקלט לבעיות מסוג זה הוא בדרך כלל אוסף של נתונים/מסמכים. הרשת עם המבנה הנוסף שלה כגרף, מאפשרת את השיפור של טכניקות קיימות עם טכניקות מעולם הגרפים. במאמר הודגם שיפור זה באמצעות פתרונות מבוססי גרפים לבעיות הבאות: חיפש נושא, מיספור נושא, סיווג וזחילה (crawling).

# סקירת מצב קיים

כיום כאשר משתמש רוצה לבצע מעקב/שינוי נתונים באתר מסויים פרושות לפניו מספר אפשרויות:

תמיכה של האתר עצמו בסוכן רשת אשר מיידע את המשתמש לגבי נתון ייחודי לאתר אליו הוא שייך. לדוגמא האתר ebay מציע סוכן שיציע במכרז מחיר עבור פריט מסויים עד תקרה שיקבע המשתמש.

מעקב מחזורי אישי של המשתמש אחר שינויים באתר.

שימוש בתוכנות אוטומציה יקרות כדוגמת QFT או RFT.

QTP(Quick Test Pro ) היא חבילת תוכנה מבית מרקורי שבה ניתן לבצע אוטומציה לאפליקציות חלונאיות ורשתיות כגון אוטומציה לכתיבת דואר אלקטרוני ושליחתו או חקירת דף רשת והכנסת נתונים אליו.

RFT (Rational Functional Tester) היא גם חבילת תוכנה לבדיקות אוטומטיות מבית IBM המאפשרת גם לבצע בדיקות על אלמנטים מסוג Flex וטרמינלים כגון VT102.

מחירם של כל אחת מחיבולות אלו מגיע לכמה אלפי דולרים עבור רשיון למשתמש אחד.

# מסמך דרישות ראשוני ותיחום של המערכת

## גורמי הפרויקט

### תכלית הפרויקט

להוות כלי למעקב, והתראה על שינויים באתרי אינטרנט וביצוע פעולות מתוכנתות מראש כתגובה למידע הנכרה מהם, באותם אתרים עצמם או אתרי מטרה אחרים.

### לקוחות הפרויקט

משתמשי דפדפן מתקדמים הרוצים לבצע פעולות אוטומטיות בדפדפן ללא מגע יד אדם.

### משתמשי הפרויקט

ראה לקוחות

## אילוצי הפרויקט

### אילוצים ראשיים/קריטיים

תמיכה ב-IE בלבד.

### הגדרות שמות, קיצורים וראשי תיבות

|  |  |  |
| --- | --- | --- |
| שם/קיצור/ר"ת | פירוש באנגלית | פירוש בעברית |
| IE | Internet Explorer | דפדפן מבית מיקרוסופט |
| HTML | Hyper Text Markup Language | שפה לתיאור טקסקט עשיר |
| DOM | Document Object Model | מודל אובייקט מסמך |
| אלמנט | Element | פריט בדף HTML |
| URL | Uniform Resource Locator | מאתר משאבים אחיד |
| Hostname | Server name part of the URL | שם השרת ב-URL |
| WWW | World Wide Web | רשת האינטרנט העולמית |

### עובדות והנחות הקשורות לפרויקט

הדף הנבדק לא יכיל אובייקטי Flash ו-Java

למשתמש יש ידע בסיסי ב-HTML

הדפדפן יהיה מסוג IE.

## דרישות פונקציונאליות

### תחום העבודה

רשת האינטרנט (WWW)/אינטרהנט.

### תחום המוצר

ניטור ואוטומציה של אתרים.

### דרישות מידע ופונקציונאליות

* בסיס נתונים לשמירת מיפוי אלמנטים בדף לאובייקטים, ולשמירת טבלת מצב-פעולה (trigger-action table)
* ממשק משתמש:
  + מיפוי אלמנטים בדף לאובייקטים בבסיס נתונים.
  + כתיבת תסריט פעולה.
  + קביעת תזמון ריצה.
* יכולת זיהוי אלמנטים בדף HTML.
* יכולת הכנסת נתונים אוטומטית בדף HTML.
* יכולת דגימה מחזורית של שדות מסויימים באתרים מסויימים
* יכולת התראה על אירוע למשתמש במצעות חלון /או דואר אלקטרוני
* המשתמש יוכל להגדיר באילו תנאים תבוצע פעולה, אותה יוכל להגדיר גם כן (לדוגמה לחיצה על כפתורים ומילוי שדות כאשר ערך של שדה מסויים ישתנה)

## דרישות לא פונקציונאליות

### דרישות עיצוב

הממשק יעוצב כך שמשתמש דפדפן ממוצע יוכל להבין בצורה אינטואיטיבית את פעולתו.

### דרישות שימוש וממשק משתמש

הממשק יהיה מעוצב כממשק לשוניות כאשר:

* הלשונית הראשונה תכיל את ממשק מיפוי האלמנטים
* הלשונית השניה תכיל את ממשק כתיבת התסריטים
* הלשונית השלישית תכיל את ממשק תזמון והרצת התסריטים.

### דרישות ביצועים

המערכת תפעל במהירות גבוהה ממהירות עבודה של משמתמש אנושי.

### דרישות הפעלה

* חיבור לרשת.
* IE מותקן.

### דרישות תחזוקה ותמיכה

תיקוני באגים ובוצעו ע"פ דרישות הלקוח.

### דרישות אבטחה

סיסמאות ופרטים אישיים יוצפנו בבסים הנתונים

### דרישות תרבותיות ופוליטיות

NA

### דרישות חוקיות

NA

## נושאי הפרוייקט

### נושאים פתוחים

* האם יוכלו לרוץ כמה תסריטים במקביל
* האם להכניס מנגנון שיתמוך בשינויים באתר/ים

### פתרונות מדף

* Newbie (<http://www.newbielabs.com/webautomation.html>)
* iMacros (<http://wareseeker.com/publisher/imacros-website-testing-and-we/38865/>(
* iRobot (<http://www.irobotsoft.com>)
* Selenium (<http://seleniumhq.org/>)
* Rational Functional Tester (<http://www.ibm.com/software/awdtools/tester/functional/index.html>)
* QTP (<http://mercuryquicktestprofessional.blogspot.com/>)

### בעיות חדשות

NA

### משימות

המשימות הנדרשות מתוארות ב-WBS ובתוכנית העבודה.

## העברת המערכת לידי/לניהול הלקוח

מוצר מדף – יתבצע באמצעות תיעוד ודוגמאות שימוש.

## סיכונים

ראה ‏13 (ניהול סיכונים)

## עלויות

פרוייקט תוכנה בלבד – ההשקעה היא זמן פיתוח בלב

## תיעוד ולימוד משתמשים

התיעוד לפרוייקט יהיה בהערות בגוף הקוד, בדוחות הביניים ובספר הפרוייקט.

## דרישות לדור הבא

לא ידוע עדיין.

## רעיונות לפתרון

ראה ניתוח פונקציונלי ראשוני לפירוט הרעיון

# ניתוח פונקציונאלי ראשוני

פעולת המערכת נחלקת לשלושה שלבים:

* איסוף/מיפוי האלמנטים ה-"מעניינים"

בשלב זה המשתמש גולש לאתר המטרה באמצעות הדפדפן המוטמע באפליקציה, ומסמן אלמנטים רלוונטיים בדף באמצעות העכבר. בלחיצה על מקש במקלדת ייטען השדה המסומן לטופס מיפוי האלמנטים באפליקציה. טופס זה מכיל מיפוי בין האלמנט ב-DOM לבין הייצוג הפנימי שלו במערכת אשר יישמש ביצירת התסריטים, וכן מידע נוסף על האלמנט כגון הטיפוס שלו, הערך הנוכחי שלו וכו'.

מיפוי המחדל עבור שם אובייקט יהיה ה-id של האלמנט, אך המשתמש יוכל לשנות זאת בטופס זה.

* הגדרת תסריט פעולה

בשלב זה המשתמש מגדיר תסריט פעולה המורכב מזוגות סדורים של תנאי קיום ופעולה (להלן יקרא זוג כזה "כלל"). תנאי הקיום יכול לבדוק ערך של אלמנט (המיוצג ע"י אובייקט) ואם הוא עומד בתנאי אז יתבצע חלק הפעולה של הכלל.

לאחר הגדרת כל כלל יתבצע מעבר בדיקה לוודא את תקינות הכללים בתסריט.

* תזמון/הרצה של תסריטים

בשלב זה בוחר המשתמש תסריט להרצה וניתנת לו האפשרות לקבוע מתי ירוץ התסריט ובאיזו מחזוריות.

מפת האלמנטים וכן התסריטים ישמרו בבסיס נתונים לצורך המשכיות (Persistence).

## תרשימי DFD

תרשים הקשר:



תרשים ‏6‑1 Context Diagram(DFD0)

מיפוי אלמנטים:



תרשים ‏6‑2 מיפוי אלמנטים

עריכת מפת אלמנטים:



תרשים ‏6‑3 עריכת מפת אלמנטים

עריכת תסריט:



תרשים ‏6‑4 עריכת תסריט

הרצת תסריט:



תרשים ‏6‑5 הרצת תסריט

# ניתוח חלופות מערכתי ראשוני

|  |  |  |  |
| --- | --- | --- | --- |
| נושא לחלופה | QTP | iMacros | RFT |
| מחיר($) | 35000-70000 | 50-500 | 6500-9000 |
| כיסוי פונקצינאלי | מושלם | חסרה שליטה בתזמון | מושלם |
| קלות שימוש | מסובך – יש ללמוד VBScript | פשוט | בינוני – יש לדעת JAVA ברמה בינונית/נמוכה. |
| Scripting | VBScript | Proprietary | JAVA |

# תאור דרך הביצוע המתוכנן

* הגדרת סכמה לבסיס הנתונים – יש להגדיר את מבנה הטבלאות כגון מבנה טבלת המיפוי ומבנה טבלאות התסריטים
* הגדרת שפת תסריט – יש להחליט על האופן בו המשתמש יתכנת תסריטים במערכת. האפשרויות הנשקלות הן שימוש באבני בניין קבועות ליצירת צעדי תסריט כך שכל צעד יורכב מבחירת מספר אלמנטים קבוע באמצעות פקדים חלונאיים. אפשרות נוספת היא כתיבת תסריט בשפת C# וניתוחו באמצעות מודול הידור תסריטים.
* חלוקת המערכת למודולים – רכיבי המערכת יחולקו למודולים הניתנים להכלה במחלקות.
* הגדרת המחלקות הנדרשות בכל מודול ומודול – כל מחלקה תכיל פונקציה לוגית בודדת.
* קידוד ובדיקת כל מודול בזה אחר זה- הקידוד יעשה בשפת C# ובדיקתו תיעשה ב-NUNIT.
* אינטגרציה של כל המודולים – בשלב זה תיבנה בפעם הראשונה כל האפליקציה כמערכת אחת.
* בדיקות – ביצוע בדיקות המערכת ווידוא שכל המטרות הושגו. כמו כן יזוהו כל הבאגים ויתועדו עבור שלב התיקון.
* תיקוני באגים ובדיקות חוזרות – ניפוי השגיאות עד לפעולה מיטבית של המערכת.
* תיעוד – כתיבת הדוחות וספר הפרוייקט. כמו כן התיעוד בתוך הקוד יעשה במהלך כתיבת הקוד.

# האמצעים/הכלים הנדרשים

**סביבת פיתוח:**

סביבת פיתוח VS2008.

דפדפן IE8

חיבור לרשת

**סביבת ריצה:**

חיבור לרשת

דפדפן IE8

.NET runtime framework 3.5

# פערי ידע שעל הסטודנט להשלים

לימוד הממשקים הנדרשים (כגון mshtml.dll, docview.dll, DOM) בעיקר מאתר MSDN

# תוצרי הפרויקט

דוח ביניים 1 – יכיל את כל אשר נדרש ע"פ ההגדרה במסמך "מבנה תוצרים – הנדסת תוכנה"

* + אבטיפוס.
  + תקציר
  + סקירה ספרותית
  + מטרות העבודה
  + סקר מצב קיים
  + ניתוח חלופות מערכתי
  + ניתוח חלופות טכנולוגיות
  + תיאור פיתרון נבחר
  + תוכנית בדיקות תוכנה
  + לוחות זמנים
  + ניהול סיכונים
  + רשימת מקורות.
  + מסמכים מצורפים SRD ,STP , SPMP. הצעת פרויקט.

דוח ביניים 2 - יכיל את כל אשר נדרש ע"פ ההגדרה במסמך "מבנה תוצרים – הנדסת תוכנה"

* + תקציר
  + מטרות העבודה
  + סקירה ספרותית
  + אלטרנטיבות לתיכון המערכת
  + הפתרון שנבחר לתיכון המערכת
  + תוכנית בדיקות תוכנה
  + לוחות זמנים מעודכנים להמשך העבודה
  + הערכת ביצוע המשימות
  + ניהול סיכונים
  + רשימת מקורות
  + מסמכים: SRD, SDD, STD, SPMP. הצעת פרויקט.
  + אפליקציה מושלמת (קובץ התקנת אפליקציה, חוברת הפעלה).

ספר פרוייקט - יכיל את כל אשר נדרש ע"פ ההגדרה במסמך "מבנה תוצרים – הנדסת תוכנה"

* + יסכם את דו"חות הביניים
  + ישלב דרישות מערכת ואפיון מערכת
  + יפנה למסמכים שכתבנו בדוחות הביניים
  + ארכיטקטורת המערכת
  + דווח בדיקות תוכנה
  + דוגמאות להפעלת המערכת
  + יתאר ביצועים עלות ואמינות
  + יתאר את הצלחת המערכת או אי ההצלחה.
  + המסמכים שנעשו במשך העבודה על הפרויקט.

מערכת

* + קבצי התקנה
  + קבצי עזרה והדרכה למשתמש
  + אבטיפוס ראשוני
  + מצגת פרויקט
  + מערכת סופית

# תכנית בדיקות ראשונית

* בניית דפי דמה המכילים מגוון רב של סוגי אלמנטים וניסוי המערכת עליהם.
* בדיקת אוטומציה של חיפוש באתר הבית של גוגל.
* בדיקת ניטור שדה באתר ושליחת מייל כאשר שדה זה עובר ערך מסויים
* בדיקת מנוע יצירת תסריטים

# ניהול סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| סיכון | סיכוי | השפעה | דרך מניעה |
| חריגה מלוחות זמנים | 50% | אי הגעה לאבני דרך בפרוייקט | ניסיון להקדים ביצוע של משימות והקדשת זמן נוסף לפרוייקט |
| כשל חומרתי בסביבת הפיתוח/ריצה | 20% | עיכוב של עד כשבועיים | גיבוי כל איטרציה במערכת ניהול קוד |
| אי יכולת לזהות אלמנטים מיוחדים בדף | 50% | אי יכולת להשתמש באלמנטים אלו – ירידה בפונקציונאליות המובטחת | עדכון סעיף האילוצים בשלב מוקדם ככל האפשר |
| דפים דינמיים (Ajax, DHTML) | 80% | שיבוש יכולת זיהוי האלמנטים | התקנת מנגנוני המתנה לאלמנטים לא קיימים. |

# תוכנית ניהול הפרויקט

## Work Breakdown Structure

## תוכנית עבודה של הפרויקט



תרשים ‏14‑1

# רשימת מקורות:

1. Jon Kleinberg. **Authoritative sources in a hyperlinked environment**. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. Extended version in Journal of the ACM 46(1999). Also appears as IBM Research Report RJ 10076, May 1997. [[pdf]](http://www.cs.cornell.edu/home/kleinber/auth.pdf)
2. Soumen Chakrabarti, Martin van den Berg, Byron Dom, **Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery**, WWW8 [[html]](http://www8.org/w8-papers/5a-search-query/crawling/index.html)
3. Steve Lawrence and Lee Giles, **Searching the World Wide Web,** Science 1998 [[pdf]](http://www.neci.nj.nec.com/~lawrence/papers/search-science98/search-science98.pdf)
4. Jeffrey Dean, Monika R. Henzinger, **Finding Related Pages in the World Wide Web**, WWW8 [[html]](http://www8.org/w8-papers/4a-search-mining/finding/finding.html)
5. Oren Zamir and Oren Etzioni, **Grouper: A Dynamic Clustering Interface to Web Search Results**, WWW8, [[html]](http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html)
6. Rajan M. Lukose and Bernardo A. Huberman, **Surfing as a Real Option** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/options.html)
7. Bernardo A. Huberman and Rajan M. Lukose, **Social Dilemmas and Internet Congestion** [[abstract]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/storms.htm)
8. Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan M. Lukose, **Strong Regularities in World Wide Web Surfing**, Nature ??? [[abstact]](http://www.parc.xerox.com/spl/groups/dynamics/abstracts/InternetEcology/surfing.html)
9. William W. Cohen, Wei Fan, **Web-Collaborative Filtering: Recommending Music By Spidering the Web**, WWW9 [[html]](http://www9.org/w9cdrom/266/266.html)
10. Mike Perkowitz Oren Etzioni, **Towards Adaptive Web Sites: Conceptual Framework and Case Study**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/towards/towards.html)
11. Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki, **Unintrusive Customization Techniques for Web Advertising**, WWW8 [[html]](http://www8.org/w8-papers/2b-customizing/unintrusive/unintrusive.html)
12. Jim Hendler, **Is there an intelligent agent in your future?** Nature, Web matters, 11 March 1999. [[html]](http://helix.nature.com/webmatters/agents/agents.html)
13. Steve Kirsch, **The future of Internet search (keynote address)**, SIGIR 1999 [[pdf]](http://www.acm.org/pubs/articles/proceedings/ir/290941/p1-kirsch/p1-kirsch.pdf)
14. G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure, Proceedings of the 8th International Computing and Combinatorics Conference (COCOON), 2002.
15. [G. Pandurangan](http://www.cs.brown.edu/people/gopal) , [P. Raghavan](http://theory.stanford.edu/people/raghavan/index.html) , and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Building Low-Diameter P2P Networks**](http://www.cs.brown.edu/research/webagent/focs-2001.pdf). Proceedings of the 42th IEEE Symp. on Foundations of Computer Science. 2001.
16. S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**The Web as a graph**](http://www.cs.brown.edu/research/webagent/pods-2000.pdf). Proceedings of the 19th ACM Symposium on Principles of Database Systems, pp 1-10, 2000.
17. Thomas Hofmann, [**Learning Probabilistic Models of the Web**](http://www.cs.brown.edu/research/webagent/Hofmann-SIGIR00.pdf), ACM SIGIR 2000
18. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and [Eli Upfal](http://www.cs.brown.edu/people/eli), [**Stochastic models for the Web graph**](http://www.cs.brown.edu/research/webagent/focs-2000.pdf). Proceedings of the 41th IEEE Symp. on Foundations of Computer Science. 2000.

1. WatiN (pronounced "what-in") aims to bring an easy way to automate Web Application Testing in .Net. Inspired by Watir, WatiN has grown into a feature rich and stable framework, offering automation of Internet Explorer and Firefox. [↑](#footnote-ref-1)