

IM with crowdsourced stickers

- Instant Messaging система с високо ниво на хоризонтална скалируемост ИЛИ p2p технология с минимално участие на централен възел
- Всеки потребител може да качи и изпрати “стикер” като предостави файл и съответни тагове, които характеризират стикера (напр. “mood”, “cats”, “despicable me”)
- Всеки потребител може да си запише получен стикер и да го използва в съобщения към трети потребители (да го препраща)
- Системата следи най-често препращаните стикери и промотира колекции от такива. Колекциите се съставят на базата на тагове (напр. “10 most popular cat stickers”)
- Авторите на най-популярните стикери получават медали (напр. звезди или ниво)
- Сигурност на съобщенията (евентуално end-to-end encryption)

Open source персонален асистент

- Чрез разпознаване на глас се разпознават основни команди (“open gmail; how many unread emails; read unread email 1”)
- Потребителите могат да асоциират фиксирани фрази с набор от предефинирани команди
- Чрез система за анализ на текст на естествен език (напр. TreeTagger) се разпознават части на речта (глаголи, съществителни)
- Може да се използва open source система + корпус за разпознаване на език (напр. CMU Sphinx, НТК, Julius, Kaldi)
- По някакъв начин се съпоставя така анализираното изречение към вече известни фрази и команди
- Самообучение и самокорекция чрез обратна връзка от потребителя

Command line fruity loops

- По модела на Improptu и подобните да се дефинира език за програмиране, програмите на който управляват софтуерен “миш пулт”
- По подобие на time line на софтуери като Fruity Loops да се миксират MIDI инструменти и предварително подготвени Waves
- Да може да се прилагат real time трансформации над Waves
- Системата да позволява синтеза да става в реално време
- Програмата да служи като декларативно описание на композицията

Command line 3D animator

- По модела на Improptu и подобните да се дефинира език за програмиране, програмите на който управляват 3D сцена в реално време

- Да може да се зареждат 3D модели и декларативно да се създават прости такива
- Да може да се анимират елементи на 3D моделите като съответните трансформации се дефинират декларативно с функции и се изпълняват в реално време

Упражнения по верификация

- Да се разработят серия от примерни програми, които да се верифицират за коректност с някоя съществуваща система за верификация на код (Dafny, Why)
- Да се подберат основни алгоритми, които се учат по дисциплините УП и СДП
- Да се дефинират задачи за самостоятелно упражнение

Уеб интерфейс на система за верификация, за доказване на теореми, или за синтезиране на програми

- По подобие на [ProofWeb](#) и да се разработи уеб интерфейс, който позволява онлайн експериментиране с някоя от съществуващите системи за верификация на програми, за интерактивно доказване на теореми, или за синтезиране на програми
- Самата система да работи на сървъра и уеб приложението да комуникира с нея чрез подходящ интерфейс, евентуално със сървърно приложение, което служи като посредник
- Да има възможност за потребителски акаунти и запазване на проекти “в облака”
- Да се контролират ресурсите, които използва системата, за да не се претовари сървъра
- Ако е по-удобно, системата да работи на виртуална машина на сървъра

Упражнения по синтезиране на програми

- Да се изследват някоя от системите за синтезиране на формално коректни програми по спецификация (Atelier B, B-Toolkit)
- Да се изберат подходящи примери за програми, които да се синтезират със системата
- Да се покаже с верификатор на програми, че синтезираните програми са коректни

Верификатор за прости свойства

- Да се напише верификатор за прости свойства на програми на популярен език за програмиране
- Примерни свойства, които да се проверяват
 - равенства за интегрални типове ($x == 3$, $c == 'A'$)

- линейни неравенства ($x > 3$, $2y + 7 < 3z$)
- съждителни логически операции (конюнкция, дизюнкция, импликация, отрицание) ($b \ \&\& \ !c$)
- Да се поддържат обикновени променливи и масиви

Анализ на алгоритъм в популярен проект с отворен код

- Със система подобна на KeY да се направи анализ на някой от алгоритмите в популярен проект с отворен код
- Да се подбере такъв метод, който е сравнително самостоятелен, т.е. може да се разглежда независимо от останалата част от системата
- Да се дефинира спецификация на алгоритъма
- Да се опише стъпка по стъпка как е направен анализа и какви са изводите от него

Cordova Приложение подобно на RunKeeper

- Измерва изминато разстояние
- Възможност за записване на преход
- Проектиране на разстоянието върху карта
- Статистики от тип
 - Най-висока скорост
 - Най-дълго изкачване
 - Брой почивки