Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

GECKA: Game Engine for Commonsense Knowledge Acquisition

Erik Cambria
School of Computer Engineering
Nanyang Technological University
cambria@ntu.edu.sg

Dheeraj Rajagopal
IHPC
A*STAR
rajagopald@ihpc.a-star.edu.sg

Kenneth Kwok
Temasek Laboratories
National University of Singapore
kenkwok@nus.edu.sg

Jose Sepulveda
Games Resource Lab
Singapore Polytechnic
sepulveda_jose@sp.edu.sg

Abstract

Commonsense knowledge representation and reasoning is key for tasks such as natural language understanding. Since common-sense consists of information that humans take for granted, however, gathering it is an extremely difficult task. The game engine for commonsense knowledge acquisition (GECKA) aims to collect common-sense from game designers through the development of serious games. GECKA merges, as never before, the potential of serious games and games with a purpose. This not only provides a platform for the acquisition of re-usable and multi-purpose knowledge, but also enables the development of games that can, apart from providing entertainment value, also teach gamers something meaningful about the world they live in.

Introduction

Games with a purpose (GWAPs) are a simple yet powerful means to collect useful information from players in a way that is entertaining for them. Over the past few years, GWAPs have sought to exploit the brainpower made available by multitudes of casual gamers to perform tasks that, despite being relatively easy for humans to complete, are rather unfeasible for machines. The key idea is to integrate tasks such as image tagging, video annotation, and text classification into games, (von Ahn 2006) producing win-win situations where people have fun while actually doing something useful. These games focus on exploiting player input to (syntax, not: both create) create both meaningful data and provide more enjoyable game experiences (Thaler et al. 2011). The problem with current GWAPs is that information gathered from them is often unrecyclable; acquired data is often applicable only to the specific stimuli encountered during gameplay. Moreover, such games often have a fairly low 'sticky factor', and are often unable to engage gamers for more than a couple of minutes.

In this work, we propose a new GWAP concept, which we call GECKA (serious game engine for common-sense knowledge acquisition), that aims to overcome the main drawbacks of traditional data-collecting games by empowering users to create their own GWAPs and by mining knowledge that is highly reusable and multi-purpose.

In particular, GECKA allows users to design compelling serious games for their peers to play and, while doing so, gather common-sense knowledge useful for intelligent applications in any field requiring in-depth knowledge of the real world, including reasoning, perception and social systems simulation (Cambria et al. 2009). Besides allowing for the acquisition of knowledge from game designers, GECKA enables players of the finished games to be educated in useful ways, all while being entertained.

The knowledge gained from GECKA is encoded in an energy-based knowledge representation (EBKR) formalism (Olsher 2013), which stores data as 'semantic atoms' that can be dynamically recombined during reasoning. The use of this noetic NLP framework allows GECKA players to conceptualize the world in their own terms, at an ideal level of semantic abstraction. Players can work with knowledge exactly as they envision it, and researchers can access data on the same level as players' thoughts, greatly enhancing the usefulness of the captured data.

The structure of the paper is as follows: the first section introduces related work in the field of human computation, the second section explains the main motivations behind the development of GECKA, the third illustrates GECKA functionalities for acquiring knowledge, the fourth presents a preliminary framework evaluation, and the last section includes concluding remarks and future directions.

Related Work

GWAPs are an example of an emerging class of games that can be considered 'human algorithms', since humans act as processing nodes for problems that computers cannot yet solve. By providing an incentive for players, GWAPs gain a large quantity of computing power that can be harnessed for multiple applications, e.g., content tagging, ontology building, and knowledge acquisition by the general public.

GWAPs for Image Tagging

GWAPs are possibly most famous for image annotation. In the 'ESP' game (von Ahn and Dabbish 2004), for example, players guess content objects or properties of random images by typing what they see when it appears on the screen. Other image annotation games include: Matchin (Hacker and von Ahn 2009), which focuses on perceived image quality by asking players to pairwise choose the picture they like better, and Phetch (von Ahn et al. 2006), a game that collects explanatory descriptions of images in order to improve Web accessibility for the visually impaired. Peekaboom (von Ahn, Liu, and Blum 2006) focuses on locating objects within images by letting a player select and reveal specific parts of an image and then challenging the other to guess the correct object name, while Squigl challenges players to spot objects in images previously annotated within the ESP Game. 'Picture This' requires players to choose from a set of images the one that best suits the given query.

Image annotation games also include those intended to help streamline the robustness of CAPTCHAs, such as Magic Bullet (Yan and Yu 2009), a team game in which players need to agree on the meaning of CAPTCHAs, and TagCaptcha (Morrison, Maillet, and Bruno 2009), where players are asked to quickly describe CAPTCHA images with single words.

GWAPs for Video and Music Annotation

Besides images, GWAPs have been used for video annotation. For example, OntoTube (Siorpaes and Hepp 2008), Yahoo's Videotaggame (van Zwol et al. 2008), and Waisd (Addis et al. 2010), are all games in which two players have to quickly agree on a set of tags for the same streaming YouTube video.

GWAPs have also been exploited to automatically tag music tracks with semantic labels. HerdIt (Barrington et al. 2009), for example, asks players to accomplish various tasks and answer quizzes related to the song they are listening to. In Tagatune (Law et al. 2007), two players listen to an audio file and describe to the other what they are hearing. Players must then decide whether or not the game has played the same soundtrack to both participants.

GWAPs for the Semantic Web

Sophisticated GWAPs have also attempted to perform complex tasks such as Web-page annotation and ontology building. Page Hunt (Ma et al. 2009), for example, is a GWAP that shows players Web pages and asks the user to guess what queries would generate those pages within the top 5 hits. Results are used to improve the Microsoft Bing search engine. The game then shows players the top five page hits for the entered keywords and rewards are granted depending on how highly-ranked the assigned Web pages are within the result set. Another example, OntoPronto (Siorpaes and Hepp 2008), is a quiz game for vocabulary building that attempts to build a large domain ontology from Wikipedia articles. Players receive random articles, which they map to the most specific appropriate class of the Proton ontology (using the *subClassOf* relationship).

Another interesting game for generating domain ontologies from open data is Guess What?! (Markotschi and Volker 2010). Given a seed concept, a player has to find the matching URI in DBpedia, Freebase and OpenCyc. The resulting labels/URIs are analyzed by simple computer-game-design tools in order to identify expressions that can be translated into logical operators, breaking down complex descriptions into small fragments. The game starts with the most general fragment and, at each round, a more specific fragment is connected to it through a logical operator, with players having to guess the concept described. Other GWAPs aim to align ontologies. Wordhunger, for example, is a Web-based application mapping WordNet synsets to Freebase. Each game round consists of a WordNet term and up to three suggested possible Freebase articles, among which players have to select the most fitting.

SpotTheLink is a two player game focusing on the alignment of random concepts from the DBpedia Ontology to the Proton upper ontology. Each player has to select Proton concepts that are either the same as, or, more specific than a randomly selected DBpedia concept. Data generated by SpotTheLink generates a SKOS mapping between the concepts of the two input ontologies. Finally, Wikiracing, Wiki Game, Wikispeedia and WikipediaMaze are games which aim to improve Wikipedia by engaging gamers in finding connections between articles by clicking links within article texts. WikipediaGame and Wikispedia focus on completing the race faster and with fewer clicks than other players. On the other hand, WikipediaMaze allows players to create races for each other and are incentivized to create and play races through the possibility of earning badges.

GWAPs for Knowledge Acquisition

One of the most interesting tasks GWAPs can be used for is common-sense knowledge acquisition from members of the general public (Chklovski 2003; Speer 2007; Cambria, Xia, and Hussain 2012). One example, Verbosity (von Ahn, Kedia, and Blum 2006), is a real time quiz game for collecting common-sense facts. In the game, two players take different roles at different times: one functions as a narrator, who has to describe a word using templates, while the other has to guess the word in the shortest time possible. FACTory Game (Lenat and Guha 1989) is a GWAP developed by Cycorp which randomly chooses facts from Cyc and presents them to players in order for them to guess whether a statement is true, false, or does not make sense. A variant of the FACTory game is the Concept Game on Facebook (Herdagdelen and Baroni 2010), which collects common-sense knowledge by proposing random assertions to users (along the lines of a slot machine) and gets them to decide whether the given assertion is meaningful or not.

Virtual Pet (Kuo et al. 2009) aims to construct a semantic network that encodes common-sense knowledge, and is built upon PPT, a popular Chinese bulletin board system accessible through a terminal interface. In this game each player owns a pet, which they take care of by asking and answer-

Figure 1: Outdoor scenario. Game designers can drag&drop objects and characters from the library and specify how these interact with each other.

ing questions. The pet acts as a stand-in for other players who then receive these questions and answers, and have to respond to or validate them.

Similar to Virtual Pet, the Rapport Game (Kuo et al. 2009) draws on player efforts in constructing a semantic network that encodes common-sense knowledge. The Rapport Game, however, is built on top of Facebook and uses direct interaction between players. Finally, the Hourglass Game (Cambria, Xia, and Hussain 2012) is a timed game that associates natural language concepts with affective labels on a hourglass-shaped emotion categorization model.

Players not only earn points in accordance with the accuracy of their associations, but also for their speed in creating affective matches. The game is able to collect new pieces of affective common-sense knowledge by randomly proposing multi-word expressions for which no affective information is known. The aggregation of this information generates a list of affective common-sense concepts, each weighted by a confidence score proportional to an inter-annotator agreement, which is therefore highly useful for opinion mining and sentiment analysis (Cambria, Olsher, and Rajagopal 2014).

GECKA Motivations

An important difference between traditional artificial intelligence (AI) systems and human intelligence is the human ability to harness common-sense knowledge gleaned from a lifetime of learning and experience to make informed decisions. This allows humans to adapt easily to novel situations where AI fails catastrophically due to a lack of situation-specific rules and generalization capabilities. Commonsense knowledge also provides background information enabling humans to successfully operate in social situations where such knowledge is typically assumed (Cambria et al. 2015).

Distributed online knowledge acquisition projects have become quite popular in the past years. Examples include: Freebase[1], NELL[2], and ProBase[3]. Other examples include the different projects associated with the Open Mind Initiative, e.g., Open Mind Common Sense (Speer 2007), Open Mind Indoor Common Sense (Gupta et al. 2004), which aims to develop intelligent mobile robots for use in home and office environments, and Open Mind Common Sentics (Cambria, Xia, and Hussain 2012), a set of GWAPs for the acquisition of affective common-sense knowledge used to enrich SenticNet[4]. Whereas previous approaches have relied on paid experts or unpaid volunteers, we put a much stronger emphasis on creating a system that is appealing to a large audience, regardless of whether or not they are interested in contributing to AI. The fundamental aim of GECKA is to transform the activity of entering knowledge into an enjoyable, interactive process as much as possible.

Most GWAPs today may be fun to play for a relatively short period of time, but players are not often keen on returning. It goes to say that GWAPs generally evidence a fairly low 'sticky factor', defined as the amount of daily active users (DAUs) of an application divided by the number of monthly active users (MAUs). While MAU on its own is the most-quoted measure of a game's size, it is only effective in describing size or reach, and not engagement. Similarly, DAU can be a very valuable metric, given that it indicates how much activity a game sees on a daily basis. However, it falls into the same trap as MAU in that it does not discriminate between player-base retention and acquisition. The single-most important metric for engagement is stickiness, i.e., DAU/MAU, which enables more accurate calculation of repeat visits and average knowledge acquired per user (AKAPU).

The key to enhancing a game's sticky factor, besides great gameplay, is the ability of an application to prompt users to reach out to their friends, e.g., via stories and pictures about their gameplay. To this end, GECKA allows users to design compelling serious games that can be made available on the App Store for their peers to play (Fig. 1). As opposed to traditional GWAPs, GECKA does not limit users to specific, often boring, tasks, but rather gives them the freedom to choose both the kind and the granularity of knowledge to be encoded, through a user-friendly and intuitive interface. This not only improves gameplay and game-stickiness, but also allows common-sense knowledge to be collected in ways that are not predictable a priori.

GECKA Key Functionalities

Not just a system for the creation of microgames, GECKA is a serious game engine that aims to give designers the means to create long adventure games to be played by others. To this end, GECKA offers functionalities typical of role-play games (RPGs), e.g., a question/answer dialogue box enabling communication and the exchange of objects (optionally tied to correct answers) between players and virtual world inhabitants, a library for enriching scenes with useful

[1]http://freebase.com

[2]http://rtw.ml.cmu.edu/rtw

[3]http://research.microsoft.com/probase

[4]http://sentic.net

and yet visually-appealing objects, backgrounds, characters, and a branching storyline for defining how different game scenes are interconnected.

Branching Story Screen

In the branching story screen, game designers place scene nodes and connect them by defining semantic conditions that specify how the player will move from a scene to another (Fig. 2).

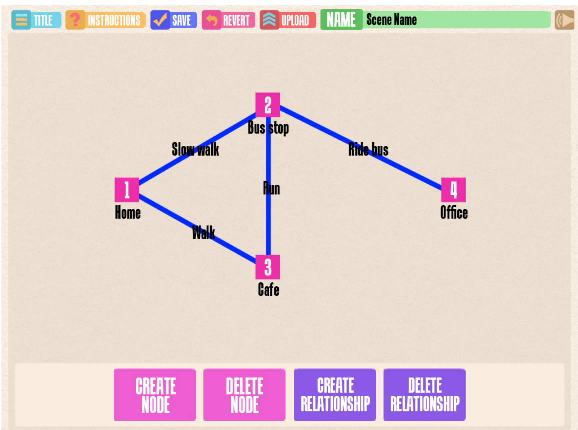

Figure 2: Branching story screen. Game designers can name and connect different scenes according to their semantics and role in the story of the game.

Making a scene transition may require fulfillment of a complex goal, acquisition of an object, or some other relevant condition. These conditions provide invaluable information about the prerequisites of certain actions and the objects that participate in action and goal flows.

In keeping with the EBKR formalism, goals are created by the combination of smaller semantic primitives ('can', 'cannot', actions, places, and so on), enabling users to specify highly nuanced goals. Designers can associate goal sequences with each story node through the combination of a set of primitives, actions, objects, and emotions (selected from the library) that describe the end state of the world once the goal sequence is complete. The branching story screen aims to acquire transitional common-sense knowledge, e.g., "if I was at the bus stop before and I am now at the office, I am likely to have taken the bus" and situational common-sense knowledge, e.g., "if you are waiting at the bus stop, your goal is probably to reach a different place".

Customizing Objects and Actions

In case an action or an object are not available in the library, GECKA allows game designers to define their own custom items by building shapes from a set of predefined geometric forms or applying transforms to existing items. This enables the creation of new objects for which there is no available icon by combining available graphics and predefined shapes, and the use of transformations to create various object states, such as a 'broken jar'. The ability of users to create their own custom items and actions is key to maintaining an undisrupted game flow.

Though the aesthetics of a custom object may not be the same as predefined icons, custom objects allow game designers to express their creativity without limiting themselves to the set of available graphics and, hence, allow researchers to discover new common-sense concepts and the semantic features associated with them.

Whenever game designers create a new object or action, they must specify its name and its semantics through prerequisite-outcome-goal (POG) triples, Prerequisites indicate what must be present or have been done before using the object or action. Outcomes include objects or states of the world (including emotional states, e.g., "if I give money to someone, their happiness is likely to rise"). Goals in turn specify the specific scene goals that are facilitated by that particular POG triple. Outcomes and Goals can be translated directly into the EBKR knowledge formalism used to store data arising from the game and used to directly support reasoning.

Defining Interaction Semantics

Game designers drag and drop objects and characters from action/object libraries into scenes. For each object, in particular, they can specify a POG triple that describes how such an object is affected by the actions performed over it (Fig. 4). POG triples give us pieces of common-sense information like "if I use a can opener on a can, I obtain the content of the can" or "the result of squeezing an orange, is orange juice".

Towards the goal of improving gameplay, and because we are mainly interested in typical common-sense knowledge, POG triples associated with a specific object type are shared among all the instances of such an object ('inheritance'). Whenever a game designer associates a POG to an object in the scene, that POG instantly becomes shared among all the other objects of the same type, no matter if these are located in different scenes. New instances inherit this POG as well. Game designers, however, can create exceptions of any object type through the creation of new custom objects. A 'moldy bread' custom object, for example, normally inherits all the POGs of 'bread' but these can be changed, modified, or removed at the time of object instantiation without affecting other 'bread' type objects.

The POG specification is among the most effective means to collect common-sense knowledge, given that it is performed quite often by the game designer during the creation of scenes. From a simple POG definition we may obtain a large amount of knowledge, including interaction semantics between different objects, prerequisites of actions, and the goals commonly associated with such actions. These pieces of common-sense knowledge, are very clearly-structured, and thus easy to assimilate into the knowledge base, due to the fixed framework for defining interaction semantics.

Defining Interaction Between Characters

POG specifications not only allow game designers to define interaction semantics between objects but also to specify

Figure 3: Status of a new character in the scene who is ill and extremely hungry, plus has very low levels of pleasantness (grief) and sensitivity (terror).

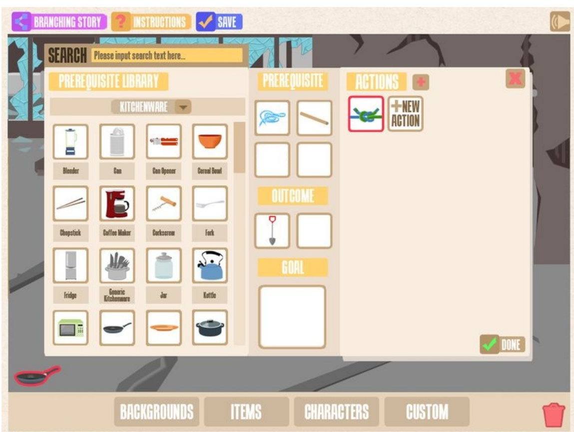

Figure 4: Specification of a POG triple. By applying the action 'tie' over a 'pan', in combination with 'stick' and 'lace', a shovel can be obtained.

how the original player, action/object recipients, and non-recipients react to various actions by setting parameters involving character health, hunger, pleasantness, and sensitivity (Fig. 3). While the first two parameters allow more physiological common-sense knowledge to be collected, pleasantness and sensitivity directly map affective common-sense knowledge onto the Hourglass model.

This is, in turn, used to enhance reasoning within the EBKR framework, especially for tasks such as emotion recognition, goal inference, and sentiment analysis. Finally, designers can also specify how characters can be engaged in a conversation by defining a question/answer dialogue and optional rewards associated with each answer.

GECKA Evaluation

In order to perform a preliminary evaluation of the type and quality of knowledge that GECKA can potentially gather, we tested it on 20 Singapore Polytechnic students, who were given the game on an iPad and were asked to design a few game scenes over the span of a few hours. Table 1 reports some of the most common POG triples collected during the pilot testing.


```xml
<scenes>
    <sceneData>
        <sceneType>
            <string>kitchen</string>
        </sceneType>
        <items>
            <itemData>
                <itemType>
                    <string>bread slices</string>
                </itemType>
                <position>
                    <x>8.04757</x>
                    <y>2.32971239</y>
                </position>
                <actions>
                    <actionData>
                        <actionType>
                            <string>stack</string>
                        </actionType>
                        <POG_Data>
                            <prerequisites>
                                <string>ham</string>
                                <string>mayonnaise</string>
                            </prerequisites>
                            <outcomes>
                                <string>sandwich</string>
                            </outcomes>
                            <goal>
                                <string>satisfy hunger</string>
                            </goal>
                        </POG_Data>
                        <player>
                            <affect>
                                <health>80</health>
                                <hunger>50</hunger>
                                <pleasantness>5</pleasantness>
                                <sensitivity>3</sensitivity>
                            </affect>
                        </player>
                        <recipientCharacter>
                            <type>hungry man</type>
                            <affect>
                                <health>80</health>
                                <hunger>50</hunger>
                                <pleasantness>5</pleasantness>
                                <sensitivity>3</sensitivity>
                            </affect>
                        </recipientCharacter>
                        <nonRecipientCharacter>
```


Figure 5: A sample XML output deriving from the creation of a scene in GECKA. Actions are collected and encoded according to their semantics.

Game designers' actions were collected and encoded according to a specific XML format that encodes the semantics associated with such actions (Fig. 5). Specific procedures translate these XML files into pieces of common-sense knowledge to be fed to the EBKR framework. Such pieces of common-sense knowledge were manually evaluated (makes sense VS does not make sense) by 5 annotators, resulting in an accuracy of 85.7%. More tests are due to verify how such new pieces of common-sense knowledge actually improve the reasoning capabilities of the EBKR framework on specific tasks such as sentiment analysis.

Conclusion

GECKA has merged, as never before, the potential of serious games and GWAPs. Future work will involve the adjustment of gameplay, incentives, and interfaces based on analysis of the data resulting from the game, leading to further understanding of the game itself as well as the nexus between game enjoyment and knowledge generation, e.g.,

Item	Action	Prereq.	Outcome	Goal
orange	squeeze	–	orange juice	quench thirst
bread	cut	knife	bread slices	–
bread slices	stack	ham, mayonnaise	sandwich	satisfy hunger
coffee beans	hit	pestle	coffee powder	–
coffee maker	fill	coffee powder, water	coffee	–
bottle	fill	water	bottled water	quench thirst
chair	hit	hammer	wood pieces	–
can	open	can opener	food	satisfy hunger
towel	cut	scissors	bandage	–
sack	fill	sand	sandbag	flood control

Table 1: List of most common POG triples collected during the pilot testing at Singapore Polytechnic.

finding out what makes providing information fun or what kinds of information are more ‘fun’ than others.

Notably, a key output of this project will be enhanced knowledge of what types of information can be gathered most enjoyably, how gathering information in specific ways enhances or detracts from playability, and how players can be incentivized and guided to provide specific kinds of information with specific semantics.

References

Addis, M.; Boch, L.; Allasia, W.; Gallo, F.; Bailer, W.; and Wright, R. 2010. 100 million hours of audiovisual content: Digital preservation and access in the PrestoPRIME project. In *Digital Preservation Interoperability Framework Symposium*.

Barrington, L.; O’Malley, D.; Turnbull, D.; and Lanckriet, G. 2009. User-centered design of a social game to tag music. In *ACM SIGKDD*, 7–10.

Cambria, E.; Hussain, A.; Havasi, C.; and Eckl, C. 2009 Common sense computing: From the society of mind to digital intuition and beyond. In *Biometric ID Management and Multimodal Communication,* volume 5707 of Lecture Notes in Computer Science, 252–259. Berlin Heidelberg: Springer.

Cambria, E.; Fu, J.; Bisio, F.; and Poria, S. 2015. AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*.

Cambria, E.; Olsher, D.; and Rajagopal, D. 2014. SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In *AAAI*, 1515–1521.

Cambria, E.; Xia, Y.; and Hussain, A. 2012. Affective common sense knowledge acquisition for sentiment analysis. In *LREC*, 3580–3585.

Chklovski, T. 2003. Learner: a system for acquiring commonsense knowledge by analogy. In *K-CAP*, 4–12.

Gupta, R.; Kochenderfer, M.; Mcguinness, D.; and Ferguson, G. 2004. Common sense data acquisition for indoor mobile robots. In *AAAI*, 605–610.

Hacker, S., and von Ahn, L. 2009. Matchin: Eliciting user preferences with an online game. In *CHI*, 1207–1216.

Herdagdelen, A., and Baroni, M. 2010. The concept game: Better commonsene knowledge extraction by combining text mining and game with a purpose. In *AAAI CSK*.

Kuo, Y.; Lee, J.; Chiang, K.; Wang, R.; Shen, E.; Chan, C.; and Hu, J. Y. 2009. Community-based game design: Experiments on social games for commonsense data collection. In *ACM SIGKDD*, 15–22.

Law, E.; von Ahn, L.; Dannenberg, R.; and Crawford, M. 2007. Tagatune: A game for music and sound annotation. In *International Conference on Music Information Retrieval*, 361–364.

Lenat, D., and Guha, R. 1989. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Boston: Addison-Wesley.

Ma, H.; Chandrasekar, R.; Quirk, C.; and Gupta, A. 2009. Page hunt: Improving search engines using human computation games. In *SIGIR*, 746–747.

Markotschi, T., and Volker, J. 2010. GuessWhat?! - Human intelligence for mining linked data. In *EKAW*.

Morrison, D.; Maillet, S.; and Bruno, E. 2009. Tagcaptcha: Annotating images with captchas. In *ACM SIGKDD*, 44–45.

Olsher, D. 2013. COGVIEW & INTELNET: Nuanced energy-based knowledge representation and integrated cognitive-conceptual framework for realistic culture, values, and concept-affected systems simulation. In *IEEE SSCI*, 82–91.

Siorpaes, K., and Hepp, M. 2008. Ontogame: Weaving the semantic web by online games. In *ESWC*, 751–766.

Speer, R. 2007. Open Mind Commons: An inquisitive approach to learning common sense. In *Workshop on Common Sense and Interactive Applications*.

Thaler, S.; Siorpaes, K.; Simperl, E.; and Hofer, C. 2011. A survey on games for knowledge acquisition. Technical report, Semantic Technology Institute.

van Zwol, R.; Garcia, L.; Ramirez, G.; Sigurbjornsson, B.; and Labad, M. 2008. Video tag game. In *WWW*.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *CHI*, 319–326.

von Ahn, L.; Ginosar, S.; Kedia, M.; Liu, R.; and Blum, M. 2006. Improving accessibility of the web with a computer game. In *CHI*, 79–82.

von Ahn, L.; Kedia, M.; and Blum, M. 2006. Verbosity: A game for collecting common sense facts. In *CHI*, 75–78.

von Ahn, L.; Liu, R.; and Blum, M. 2006. Peekaboom: A game for locating objects in images. In *CHI*, 55–64.

von Ahn, L. 2006. Games with a purpose. *IEEE Computer Magazine* 6:92–94.

Yan, J., and Yu, S.-Y. 2009. Magic bullet: a dual-purpose computer game. In *ACM SIGKDD*, 32–33.