

# 1 Executive Summary

**AxeSki** is een visuele programmeer omgeving. Het hoofddoel van de IDE is om event-based programma's te kunnen schrijven. Een gebruiker kan standaard programma's maken net zoals in andere programmeertalen. Een bijkomend voordeel is dat de gebruiker goed leert hoe event-based programming werkt. De IDE stelt dit concept eenvoudig voor zodat zelfs nieuwe programmeurs snel programma's kunnen schrijven. AxeSki biedt geen restricties, zoals een gelimiteerd aantal processen en laat gebruikers toe complexe programma's in elkaar te steken.

De gebruiker werkt voornamelijk met **Events**, dit zijn informatiepakketten die in een programma rondgezonden kunnen worden tussen verschillende entiteiten. Deze events worden aangeduid door een unieke naam. De informatie die erin zit wordt ook aangeduid door een unieke naam binnen het event.

Vervolgens kan de gebruiker met behulp van programmeerstructuren een **klasse** opbouwen. Een klasse is een entiteit binnen het programma die kan reageren op bepaalde Events, en er ook zelf kan uitsturen. Deze programmeerstructuren omvatten de standaard programmeer constructies zoals een while-loop, if-condities, operaties, variabelen, etc. Tijdens het aanmaken van de klasse moet de gebruiker niet weten waar de binnenkomende events vandaan komen, maar wel hoe deze events afgehandeld moeten worden. Een klasse kan ook terug events verzenden. Opnieuw moet de gebruiker zich niet bezighouden naar wie de events verzonden worden, maar indien nodig wel met welke informatie deze events verzonden worden.

De blokken waarmee de gebruiker het programma opbouwt, hebben een neutrale kleur. De verschillende vensters, het klasse-view, het frame-view en het event-view zijn eenvoudig in gebruik. Hierdoor heeft AxeSki een **professionele look**. Om bruikbaarheid van de IDE te bevorderen kan de gebruiker de taal aanpassen. Er zijn twee ondersteunde talen, namelijk Engels en Nederlands.

Deze klassen kunnen geïntanceerd worden en kunnen verbonden worden met elkaar om de events door te sturen. Een instantie is een black box, de gebruiker moet zich, wanneer hij instanties maakt, niet bezighouden met wat deze black box doet, maar wel hoe verschillende instanties met elkaar communiceren. De soorten Events die een instantie kan opvangen en versturen zijn deze die gespecificeerd zijn in de bijhorende klasse.

Om een programma op te bouwen beschikt de gebruiker over verschillende tools. De gebruiker kan ten alle tijde een programma **opslaan** zodanig dat zijn voortgang behouden blijft. Deze opslag gebeurt in een leesbaar formaat (XML) zodanig dat de gebruiker zelf deze bestanden zou kunnen aanpassen. In AxeSki is een editor ingebouwd om het opslagbestand te bewerken. De gebruiker heeft dus de keuze om te werken met de visuele IDE, of om het ruwe bestand aan te

passen.

Een andere beschikbare tool is **debugging**. De gebruiker kan stap voor stap door het programma heen lopen. Zodanig kan de gebruiker gemakkelijk kijken of het programma de gewenste uitvoer heeft. De gebruiker kan ook een stop-punt definiëren in het programma zodat de uitvoer zal pauzeren op dat punt. Hierdoor kan de gebruiker eventuele fouten in een programma gemakkelijker opsporen.

Een andere hulpvolle tool is **typechecking**. De gebruiker kan in zijn programma variabelen aanmaken die informatie bevatten. Deze informatie behoort tot een bepaald type, zoals een getal of string. AxeSki typechecking zorgt ervoor dat de gebruiker geen operaties tracht uit te voeren met een niet-compatiebele variabele. De ondersteunde types zijn numerieke waarden, booleaanse waarden en strings.

Om de IDE interactief te houden is een **canvas** geïmplementeerd. Instanties van klassen worden hierin getoond en kunnen tijdens het runnen van een programma hier events opvangen van de gebruiker, zoals toetsaanslagen of een muisklik. Een klasse kan ook kostuums bevatten. Dit laat toe om een instantie te koppelen aan een visuele afbeelding. Dit laat toe om de programma's die de gebruiker kan maken meer mogelijkheden tot feedback te geven.

Achter de IDE zitten verschillende complexe design patronen verwerkt. AxeSki simuleert parallele uitvoering waardoor alle instanties en events die opgevangen worden simultaan uitgevoerd. Dit gebeurt door threading te simuleren, de uitvoering wordt na een stap doorgegeven aan de volgende thread die uitgevoerd moeten worden. Echte systeem threads gebruiken is veel te zwaar voor AxeSki. De IDE moet vele threads kunnen runnen en zal dus veel context switches hebben, om het snel te houden hebben we zelf threads geïmplementeerd. Het bewegen van de blokken (de drag-and-drop) is zelf geïmplementeerd. Het correct nesten en verslepen gebeurt door zelf-geïmplementeerde logica. Er moet communicatie bestaan tussen de blokken om typechecking te kunnen weergeven. Zo kunnen hoge niveau blokken zoals functies zijn "kind" blokken updaten, en de kind blokken kunnen hun ouder laten weten dat ze aangepast zijn.

Alhoewel AxeSki functioneel is en meer dan de geplande features bevat, is er toch nog ruimte voor uitbreidingen. De IDE is zo gebouwd zodanig dat aanpassingen en toevoegingen eenvoudig te implementeren. Alle onderdelen, zoals de uitvoering en het visuele uitzicht zijn volledig losgekoppeld.

Dit project heeft ons veel geleerd. We kunnen beter omgaan met grotere software projecten. De belangrijkheid van documentatie is nog duidelijker geworden. Grote projecten vereisen goede documentatie zodat alle bouwstenen van het project correct in elkaar gestoken kunnen worden.